Relevance-Based Language Modelling for Recommender Systems

Javier Parapar[a,*], Alejandro Bellogín[b], Pablo Castells[b], Álvaro Barreiro[a]

[a] *Information Retrieval Lab, Department of Computer Science, University of A Coruña, Campus de Elviña, 15071 A Coruña, Spain*
[b] *Information Retrieval Group, Department of Computer Science, Universidad Autónoma de Madrid, Ciudad universitaria de Cantoblanco, 28049 Madrid, Spain*

Abstract

Relevance-Based Language Models, commonly known as Relevance Models, are successful approaches to explicitly introduce the concept of relevance in the statistical Language Modelling framework of Information Retrieval. These models achieve state-of-the-art retrieval performance in the pseudo relevance feedback task. On the other hand, the field of Recommender Systems is a fertile research area where users are provided with personalised recommendations in several applications. In this paper, we propose an adaptation of the Relevance Modelling framework to effectively suggest recommendations to a user. We also propose a probabilistic clustering technique to perform the neighbour selection process as a way to achieve a better approximation of the set of relevant items in the pseudo relevance feedback process. These techniques, although well known in the Information Retrieval field, have not been applied yet to recommender systems, and, as the empirical evaluation results show, both proposals outperform individually several baseline methods. Furthermore, by combining both approaches even larger effectiveness improvements are achieved.

Keywords:
Relevance Models, Recommender Systems, Collaborative Filtering, Probabilistic Clustering

1. Introduction and Motivation

Recommender Systems have traditionally been a fertile research area due to the existence of a wide range of scenarios where users may benefit from automatic personalised recommendations. This research area has its roots in the eighties, and started to attract wider attention in the mid-nineties when the

*Corresponding author. Tel.: +34 981 167 000x1276; Fax: +34 981 167 160.
Email addresses: javierparapar@udc.es (Javier Parapar), alejandro.bellogin@uam.es (Alejandro Bellogín), pablo.castells@uam.es (Pablo Castells), barreiro@udc.es (Álvaro Barreiro)

first works on collaborative filtering were published (Resnick, Iacovou, Suchak, Bergstrom, and Riedl, 1994; Hill, Stead, Rosenstein, and Furnas, 1995). Collaborative Filtering (CF) is one of the three classical approaches to recommendation (Adomavicius and Tuzhilin, 2005): *content-based recommendation*, based on the user's history; *collaborative filtering*, based on the history of similar users; and *hybrid approaches*, based on combining content-based recommendation and collaborative filtering.

In CF (Herlocker, Konstan, and Riedl, 2002), the input evidence about user preferences consists of data records collected from user interaction with items. In the simplest form, this evidence consists of explicit user ratings, which are graded relevance values assigned by end-users to items of interest. CF algorithms exploit the target user's ratings to make preference predictions, and have the interesting property that no item descriptions are needed to provide recommendations, since the algorithms merely exploit information about past interaction between users and items. Moreover, CF has the salient advantage that a user benefits from others' experience, being exposed to novel recommendations produced from the personal preferences of affine users.

Two different types of CF approaches exist: model-based approaches, which learn user/item rating patterns to build statistical models that provide rating estimations, and memory-based approaches, which compute user/item similarities based on distance and correlation metrics (Desrosiers and Karypis, 2011). Memory-based approaches find either like-minded people for the target user (user-based approach), or pairs of items that are liked by common users. In the user-based approach, the set of similar-minded users are called neighbours, and their preferences are combined to predict ratings for the active user. In the item-based approach, items similar to the ones the user has liked in the past are recommended.

The recommendation task has been traditionally formulated and evaluated as a rating prediction problem (Adomavicius and Tuzhilin, 2005). However, in practical terms, the effectiveness of recommendations depends on what items are presented to the user and in what order. Thus the ranking of recommended items, rather than the numeric system scores that determine this ranking, is the essential problem in common recommendation scenarios, whereby recommendation can be seen as an IR task – one where there is no explicit query. Considering this, several proposals have been recently developed to formalise and address the recommendation task as a relevance ranking problem (Wang, de Vries, and Reinders, 2006a, 2008a; Wang, Robertson, de Vries, and Reinders, 2008b; Bellogín, Wang, and Castells, 2011c). The objective is to take advantage of well-studied and highly-performing Information Retrieval (IR) techniques to achieve effectiveness enhancements – and a better theoretical understanding – in recommendation tasks, upon principles of ranking for relevance. Authors in this strand of research have explored the adaptation of the vector-space IR model (Bellogín et al., 2011c), the extended Boolean model (Bellogín, Wang, and Castells, 2011b), the binary independence retrieval model (upon the probability ranking principle) (Wang et al., 2008a,b), and statistical Language Models (Wang et al., 2006a). However, to the best of our knowledge, no attempt has

been made yet at a similar adaptation of so-called Relevance-Based Language Models (Lavrenko and Croft, 2001).

Relevance-Based Language Models (or Relevance Models for short, RM) are among the best-performing ranking techniques in text retrieval. They were devised with the aim of explicitly introducing the concept of relevance, intrinsic to the probabilistic model of IR, in statistical Language Models (LM). Relevance Models achieve state-of-the-art performance in terms of effectiveness for the pseudo relevance feedback task (a.k.a. blind relevance feedback) (Lavrenko and Croft, 2001). In IR, relevance is a relation between a query and a set of documents. In common IR settings, the exact and complete set of relevant documents is generally unknown. Relevance feedback techniques work with approximations to this set, which can be obtained by a wide variety of approaches, such as asking the user (explicit relevance feedback), or just taking the initial output of a well-performing IR system as a good guess (pseudo relevance feedback) – this is the case in RM. Given a query and such an approximation to the set of relevant documents, RM selects good expansion terms from those present in the pseudo-relevant documents in order to formulate and run a better query.

The adaptation of RM to recommendation is non-trivial as, to begin with, there are neither queries nor words in the generic recommendation task. A first problem we therefore address is to find an analogy between the elements involved in RM as defined in text retrieval (documents, queries, words, pseudo-relevant documents, expanded terms), and the variables handled by a recommender system: users, (black-box) items, and records of interaction between them. In this paper we propose one such analogy under which the RM can be adapted to recommendation, leveraging the effectiveness of the Relevance Models to estimate the probabilities of relevance, even when the probability distributions are not expressed in terms of words as originally proposed for text retrieval. Our approach involves, as we shall see, the selection of similar users as the equivalent of pseudo-relevant documents, resulting in a form of user profile expansion through the preferences of nearest neighbours.

A good approximation of the set of relevant documents is critical to the effectiveness of pseudo relevance feedback methods. Analogously, a good selection of user neighbourhoods (as the equivalent of pseudo-relevant documents) can be expected to heavily influence the effectiveness of our approach. In the context of a probabilistically formalised framework as we intend to build, we investigate the use of Posterior Probabilistic Clustering (PPC) (Ding, Li, Luo, and Peng, 2008) as a rigorous probabilistic basis for neighbourhood formation, based on Non-negative Matrix Factorisation (NMF). Besides the probabilistic interpretability of this method, the NMF family of algorithms has proved to have a very good performance in terms of clustering effectiveness (Xu, Liu, and Gong, 2003). In this paper we explore the novel use of this particular probabilistic clustering in recommender systems, both in isolation (as an enhancement of neighbour selection in CF recommendation), and in combination with the relevance modelling of the recommendation process.

The main contributions of this paper thus include:

- A new recommendation approach based on Relevance Modelling under the Statistical Language Modelling framework, where the recommendation process is modelled as a profile expansion process. We produce new estimations for RM under the i.i.d. sampling and conditional sampling assumptions in Recommender Systems. We develop the probabilistic framework into computable terms, resulting in a novel, empirically effective recommendation method.

- The use of probabilistic clustering methods for the neighbour selection problem, in particular, the use of Posterior Probabilistic Clustering, for which we have produced the necessary document representation strategies to be able to use PPC for a task so different from that originally conceived (text clustering).

- The combination of both contributions, further enhancing the performance of their separate application. We find performance improvements of over 300% with respect to the best method tested from the state-of-the-art.

The remainder of the paper is structured as follows: in Section 2 we present a study of the works related to our proposal. Section 3 presents the Relevance Modelling framework and its adaptation to the recommendation problem. In Section 4 we introduce our proposal for neighbour selection based on Posterior Probabilistic Clustering. Section 5 reports the empirical evaluation of the proposed approaches and analyses the results of different experiments. Finally, conclusions and future work directions are presented in Section 6.

2. Related Work

The use of probabilistic modelling from Information Retrieval in Collaborative Filtering has been research before by several authors. In (Wang et al., 2008b), the authors found interesting analogies between CF with implicit data (where the evidence of user interest for items consists of access frequencies, rather than explicit preference rating values) and IR, introducing the concept of binary relevance into CF and applying the Probability Ranking Principle of IR to CF. Similarly, in (Wang et al., 2006a) a generative relevance model is proposed for implicit CF, and in (Wang, 2009), the author made use of a language modelling formulation to propose a risk-aware ranking for implicit CF. The approach we propose here is much in tune with the spirit of this line of research on model unification. A fundamental difference is that the aforementioned related work is restricted to recommendation settings where the user activity records provide frequency of (repeated) user interaction with items, and cannot be applied to explicit rating data – a very common source of groundtruth data in CF – as our approach does. Furthermore, as far as we know our approach is the first that fully adapts the Relevance Models as proposed in (Lavrenko and Croft, 2001).

Regarding rating-based CF, one of the first works which explicitly dealt with a generative probabilistic framework in a rating-based collaborative filtering scenario is (Wang, de Vries, and Reinders, 2006b). In that work, together with (Wang et al., 2008a), the authors presented a probabilistic relevance framework, where three models are derived: one based on users, another based on items, and a unified relevance model. This modelling approach was based on the probabilistic interpretation of the Relevance Models for language modelling (Lafferty and Zhai, 2002). We share with this work the goal to model the recommendation problem as a relevance ranking task. The key difference is that Wang et al. seek the unification of probabilistic IR and recommendation principles in producing an initial ranking, while our proposal unifies query expansion and (a novel notion of) user profile expansion in the recommendation process. Furthermore, a comparative empirical evaluation of both proposals shows that our methods achieve better results in terms of precision-based metrics, whereas Wang's methods perform well with error-based metrics such as Mean Absolute Error. Furthermore, the relevance models by Wang et al. require a considerably involved and expensive training phase (based on Expectation Maximisation) to learn the optimal parameter values (bandwidth h_u in (Wang et al., 2008a)), while the methods proposed herein only have two simple parameters (linear smoothing coefficient and number of clusters, as we shall see) for which very simple tuning approaches are sufficient, and even default values from other collections achieve a decent performance, as we shall show in our experiments.

Regarding the use of clustering for recommendation, some authors split the set of users or items in order to improve the scalability of the recommender systems and their accuracy (O'Connor and Herlocker, 1999; Xue, Lin, Yang, Xi, Zeng, Yu, and Chen, 2005). Most of these approaches use classic clustering methods such as k-Means or hierarchical clustering, which, in general, produce good results but at the expense of lower coverage (Xue et al., 2005). Furthermore, some approaches require external or additional information, such as the content data of the item (e.g. genres or tags, in the movie domain). In this line of research, we have recently studied the use of the Normalised-Cut algorithm for neighbour selection in (Bellogín and Parapar, 2012), showing important improvements over classical clustering approaches but not reaching the performance of the presented PPC-based approach.

Although there are no Posterior Probabilistic Clustering applications in the field of recommendation, Non-negative Matrix Factorisation (NMF) methods have been previously used, mainly for the rating prediction task, as a model-based recommender similar to, for instance, SVD. In Gu, Zhou, and Ding (2010), a unified model is proposed for collaborative filtering based on a type of non-negative matrix factorisation algorithm. While, in our case, Posterior Probabilistic Clustering is only a better performing tool for locating good neighbourhoods, in that work the NMF algorithm also produces recommendations itself, by combining both model-based and memory-based information to improve the recommendation effectiveness. The evaluation against existing methods exhibited however modest improvements in terms of Mean Absolute Error. Among other baselines, they compared the results with a previous work

by (Zhang, Wang, Ford, and Makedon, 2006) that also uses different types of NMF algorithms. The latter was a pioneering work on tackling the problem of incomplete ratings when applying recommendations based on weighted NMF, obtaining small improvements against user-based and matrix factorisation techniques, again in terms of Mean Absolute Error.

3. Relevance-Based Language Modelling for Recommendation

We will first briefly review the Relevance Models in their original context, after which we shall present our proposal for the adaptation of relevance modelling to the recommendation problem.

3.1. Relevance Models in Information Retrieval

Blind feedback or pseudo relevance feedback is a local query expansion technique used for improving retrieval effectiveness. The basic assumption in pseudo relevance feedback is that a high number of top documents initially returned by a retrieval system are relevant. Given that assumption, the idea is to choose from those documents good terms to expand the original query in order to improve the effectiveness in a second document ranking with the expanded version of the query.

In (Lavrenko and Croft, 2001), Relevance Models were presented under the Language Modelling framework and proved to be very successful to improve retrieval effectiveness. Relevance Models have not only been proposed as an effective method for pseudo relevance feedback but also as a robust one (Lv and Zhai, 2009). Moreover, Relevance Modelling has already been used for other tasks and in combination with other approaches such as the employment of query variants (Collins-Thompson and Callan, 2007), sentence retrieval (Balasubramanian, Allan, and Croft, 2007), cluster based retrieval (Lee, Croft, and Allan, 2008), passage retrieval (Li and Zhu, 2008), constrained text clustering (Parapar and Barreiro, 2012), etc.

In RM, the original query Q is seen as a short sample of words obtained from the relevance model (R_Q). If more words from R_Q are desired then it is reasonable to choose those words with the highest estimated probability when considering the words for the distribution already seen. So the terms in the lexicon of the collection are sorted according to that estimated probability. Lavrenko and Croft (2001) originally presented two different estimations for RM, namely RM1 and RM2.

First, in RM1 we assume that the query words $q_i \in Q$ and the words w in the relevant documents are sampled identically and independently from a unigram distribution (*i.i.d. sampling*), thus, the probability $p(w|R_Q)$ is computed as in Eq. 1 (Lavrenko and Croft, 2001):

$$p(w|R_Q) \propto \sum_{d \in \mathcal{C}} p(d)p(w|d) \prod_{i=1}^{|Q|} p(q_i|d) \tag{1}$$

where $\mathcal{C}$ is the set of all documents in the search space (the collection).

On the other hand, in RM2 (*conditional sampling*) the main assumption is that the query words are independent from each other but dependent on the words of the relevant documents. As a result of that, $p(w|R_Q)$ is computed as in Eq. 2 (Lavrenko and Croft, 2001)[1]:

$$p(w|R_Q) \propto p(w) \prod_{i=1}^{|Q|} \sum_{d \in \mathcal{C}} p(q_i|d) \frac{p(w|d)p(d)}{p(w)} \tag{2}$$

The final ranking is obtained in four steps:

1. Initially, the documents in the collection $\mathcal{C}$ are ranked under the LM framework using the query likelihood retrieval function. This query likelihood is usually estimated with some form of smoothing.
2. The top N documents from the initial result set are taken for the estimation, instead of the whole collection $\mathcal{C}$. Let us call this pseudo relevance set or RS.
3. The relevance model probabilities $p(w|R_Q)$ are calculated using the estimation presented in Eq. 1 or Eq. 2.
4. To build the expanded query, the e terms with highest estimated probabilities $p(w|R_Q)$ are selected. The expanded query is used to produce a second document ranking using the negative cross entropy retrieval function, as follows

$$\text{score}(d, q) = \sum_{w} p(w|R_Q) \log p(w|d) \tag{3}$$

3.2. Relevance Modelling for Recommendation

As we have seen in the previous section, RM in text retrieval operates on three main spaces: words, documents, and queries. Words are related to queries and documents by direct observation, which enable the estimation of conditional probabilistic relations between the three sampling spaces. Documents play two different roles: as objects to be ranked, and as pseudo-relevant objects. So do words, as elements defining the initial information need expression, and as terms for query expansion. Yet the general recommendation task considers just two tangible fundamental variables: users and items, and the direct observations only involve the interaction between them (e.g. users assign rating values to items). Some probabilistic approaches to recommendation consider rating values as objects in their own right (Billsus and Pazzani, 1998), but we do not consider this option here; we rather see ratings as an intrinsic property of the relation between users and items which is equivalent to the role of the term weights in the RM formulation for retrieval.

[1]Please, note that in the original paper there is an erratum in the step of applying Bayes' rule for RM2 estimation, therefore the equations here and there are different, although consistent with subsequent works such as Lv and Zhai (2009)

The adaptation of RM to recommendation thus involves a non-trivial transition from a triadic space (queries, documents, words) to a dyadic one (users and items). We propose to achieve this mapping as follows: first, the role of the query in the retrieval task is played by the user to whom we want to provide with item recommendations (target user). Now, as stated before, the objective in pseudo relevance feedback is to select from the pseudo relevant set good terms which are related to the original query terms. In the case of retrieval, the goodness of those selected terms is evaluated by how adding them to the original query produces a more effective second document ranking. In recommendation and, particularly, in collaborative filtering, the user is modelled as a set of previously scored items. In our approach, we propose to have those items play the role of the query words in IR relevance models. This is how in our adaptation of RM to recommendation, query expansion translates to a form of user profile expansion, where the objective is to expand the user need representation, embodied in the user profile, with further items related to her expressed interests. In this formulation, the recommendation process becomes a profile expansion problem, where items to be recommended play the role of the candidate expansion terms in the pseudo relevance feedback task. Finally, the role of the pseudo-relevant documents is played in our model by the set of similar users (the user neighbourhood, based on profile similarity with the target user). The same as the selection of suitable pseudo-relevant documents is a critical aspect in text retrieval, the selection of suitable user neighbours is key to the effectiveness of our proposed adaptation, which motivates the research of adequate neighbour formation techniques. We will assume for the moment the neighbourhood is given, and we will describe later in Section 4 how we address this part of the problem by a probabilistic clustering approach.

The set of analogies on which our approach is based is shown in Figure 1. The reader may have expected to find in our explanation a classical and natural equivalence between retrieved text documents and recommended items. This is intentionally missing in our approach because, differently from the application of RM in text retrieval, we propose to take the result of profile expansion itself as the final recommendation output, skipping the final, second ranking step in RM (Eq. 3). That is, the output of the resulting recommendation algorithm is the equivalent of the output from term selection in RM (Eq. 1 and 2). Since the elements for expansion are items already (belonging to the final retrieval space), we omit the subsequent re-ranking which in RM uses the selected terms as input (Eq. 3).

The triadic/dyadic problem is thus addressed in our framework by:

- Having users – or to be more precise, user profiles – play a dual role, as a) a representation of user needs, equivalent to a query, (target user) and b) a pseudo-relevant document, serving as a source for item selection in profile expansion (target user's neighbours).
- Omitting the retrieved document variable from our formulation, by equating expansion terms and retrieved items, and skipping the second ranking in RM.

Figure 1: Analogies between the use of Relevance-Based Language Models for document retrieval and for item recommendation

Relevance Models for Text Retrieval	Relevance Models for Recommendation
query (Q)	target user (u)
query words ($q_1 \dots q_n$)	items rated by user ($I(u)$)
pseudo relevance set (RS)	user neighbourhood (V)
candidate terms for query expansion	candidate items for user recommendation

The recommendation problem can be thus accommodated as an expansion process where the models for pseudo relevance feedback can be tested. In order to accomplish the proposed adaptation, we need to assume that for every target user $u \in \mathcal{C}$ and set of *relevant users* or *neighbours* (V), an underlying Relevance Model R_u exists. This supporting relevance model can be estimated under the RM framework and, from this estimation, the ranking of best items to recommend to the user u are selected. It is important to note that this model is agnostic with respect to how the relevant users are determined, that is, different neighbour selection methods can be incorporated in a straightforward way. Indeed, we will go back to this point later on and show how different selection approaches can be integrated into our model.

Using neighbours as pseudo-relevant documents has an additional consequence, which we would like to stress: differently from RM in text IR, our approach does entirely without an initial, first-step ranking – the ranking RM would require in text IR to select the top N documents as the pseudo-relevant set.

In the following two subsections we show how the two models RM1 and RM2 proposed in the IR field (Lavrenko and Croft, 2001) can be effectively adapted to the recommendation task, following the proposed analogies.

3.2.1. Method 1: i.i.d. sampling

Analogously to the RM1 estimation in IR, we produce RM1-based recommendations. In this context, we assume that the items in the user's profile and the items rated by the user's neighbours are sampled identically and independently from a unigram distribution. Eq. 4 defines the estimation of probabilities in the Relevance Model underlying u and V. For every item i in the set of items scored by the similar users V (where V acts as the relevance set) the probability of the item i given the relevance model R_u for user u is computed as:

$$p(i|R_u) \propto \sum_{v \in V} p(v)p(i|v) \prod_{j \in I(u)} p(j|v) \tag{4}$$

where $I(u)$ is the set of items already rated by the user u.

Therefore, $\prod_{j \in I(u)} p(j|v)$ being the user's profile likelihood for the neighbour v and assuming the prior for a user's neighbour as uniform, we can estimate the probability of an item under the Relevance Model for a given user, as the

weighted average of the language model probabilities for the item in the neighbourhood of the user, where the weights are the user profile likelihood scores for her neighbours.

Given this scoring formula, the top items can be selected for recommendation by ranking the items according to the probability $p(i|R_u)$. The reader may note the equivalences between this equation and Eq. 1 for text retrieval, where items and users have a dual role: for items, retrieved items correspond to expansion terms $i \rightarrow w$ and the items rated by the user correspond to query terms $j \in I(u) \rightarrow q_i$, for users, the target user is the interpreted as the query $u \rightarrow Q$ and the neighbours as pseudo-relevant documents $V \rightarrow RS$.

3.2.2. Method 2: conditional sampling

Alternatively, we can make use of the conditional sampling assumption as in the RM2 method. In this case, we assume that items in the user's profile are independent from each other but dependent on the items present in the profiles of the user's neighbours. In this situation, the item preference is computed as follows:

$$p(i|R_u) \propto p(i) \prod_{j \in I(u)} \sum_{v \in V} p(v|i)p(j|v) \tag{5}$$

where $p(v|i)$ is estimated with Bayes as $p(i|v)p(v)/p(i)$, that is, the preference score is:

$$p(i|R_u) \propto p(i) \prod_{j \in I(u)} \sum_{v \in V} \frac{p(i|v)p(v)}{p(i)} p(j|v) \tag{6}$$

Therefore, as Eq. 5 shows, in this case the association between each item and the user's profile is computed using the neighbours that contain both the profile's items and the item as "bridges". Apart from that, the analogy to Eq. 2 for text retrieval is the same as the correspondence between Eq. 1 and 4 for RM1 underlined in the previous sections.

3.2.3. Final Estimation Details

For both methods we can initially consider that the prior $p(v)$ is uniform, i.e. every neighbour ($v \in V$) has the same probability of being sampled. The estimation of the probability of an item given a user will be computed by smoothing the maximum likelihood estimate with the probability in the collection (background collection model), in this case using Jelinek-Mercer smoothing (Zhai and Lafferty, 2004):

$$p_\lambda(i|u) = (1 - \lambda)p_{ml}(i|u) + \lambda p(i|\mathcal{C}) \tag{7}$$

where $I(\mathcal{C})$ is the set of items in the collection and $p_{ml}(i|u)$ is estimated as:

$$p_{ml}(i|u) = \frac{\text{rat}(u, i)}{\sum_{j \in I(u)} \text{rat}(u, j)} \tag{8}$$

in this case $\text{rat}(u, i)$ represents the rating assigned by user u to item i, and $p(i|\mathcal{C})$ is estimated as a maximum likelihood in the whole collection:

$$p(i|\mathcal{C}) = \frac{\sum_{v \in \mathcal{C}} \text{rat}(v, i)}{\sum_{j \in I(\mathcal{C}), v \in \mathcal{C}} \text{rat}(v, j)} \tag{9}$$

In the Language Modelling framework and retrieval tasks, Dirichlet smoothing outperforms Jelinek-Mercer (Zhai and Lafferty, 2004). However, when modelling the recommendation problem, Dirichlet can suffer from the undesired effect of demoting those items that have been recently introduced in the system and so have very few recommendations. In fact, in (Wang, 2009) the smoothing for the LM based recommendation with Dirichlet smoothing presents significantly worse performance than using Jelinek-Mercer in one of the experiments reported there. For estimating $p(i)$ we decided to keep it simple and a uniform distribution was chosen.

Finally, depending on the proposed methods, different strategies were used in this paper to compute the neighbourhood of a given user (V), as we present in the next section.

4. A Probabilistic Neighbour Selection Technique

A crucial step in order to rank the items according to the RM framework is to properly select the relevance set. In our adaptation of the RM framework to user-based collaborative filtering, this relevance set is composed by the target user's neighbourhood, that is, the set of her most akin users. Next, we will define an alternative probabilistic approach to the computation of such neighbourhoods. This is not enforced by the RM approach itself, and other alternatives could thus be considered as well, which we leave as future work. A probabilistic neighbour selection approach provides nonetheless for a smoother global user-based CF framework. In particular, the approach proposed here builds on the Posterior Probabilistic Clustering algorithm, as we present next.

4.1. Posterior Probabilistic Clustering (PPC)

The lack of probabilistic interpretation of Non-negative Matrix Factorisation (NMF) (Lee and Seung, 2001) clustering methods and their ad-hoc document-to-cluster assignments motivated the development of the Posterior Probabilistic Clustering (PPC) method (Ding et al., 2008). PPC provides with a posterior probability interpretation, removes uncertainty in the clustering assignment and has a very close relation to probabilistic latent semantic indexing when performing co-clustering of documents and words.

Given a collection of n documents and m words, let $X = (X_{ij})$ be the words-to-documents matrix where $X_{ij} = X(w_i, d_j)$ is the term frequency of the term w_i in the document d_j. The traditional formulation of the NMF method consists in solving the following optimisation problem, given a number of clusters κ:

$$\min_{F \geq 0, G \geq 0} \| X_{m \times n} - F_{m \times \kappa} G^T_{\kappa \times n} \|^2 \tag{10}$$

Once the solution (G^*, F^*) to the optimisation problem is obtained, every document d_j is assigned to the cluster C_κ such that:

$$\kappa = \arg\max_z(G^*_{jz}) \tag{11}$$

where z ranges from 1 to κ.

PPC is a posterior probability interpretation of the NMF algorithm. PPC considers the rows of G^* as the posterior probabilities that a given document belongs to the different clusters, i.e. $p(d_j|C_l) = G^*_{jl}$. In order to enforce a proper probability distribution, a PPC optimisation function is formulated as follows:

$$\underset{F\geq 0, G\geq 0}{min} \parallel X_{m\times n} - F_{m\times\kappa}G^T_{\kappa\times n} \parallel^2, \quad s.t. \sum_{k=1}^{\kappa} Gj_\kappa = 1 \tag{12}$$

which results, after using Lagrangian multipliers, in the next updating rules:

$$G_{i\kappa} \longleftarrow G_{i\kappa}\frac{(X^TF)_{i\kappa} + (GF^TFG^T)_{ii}}{(GF^TF)_{i\kappa} + (X^TFG^T)_{ii}} \tag{13}$$

$$F_{i\kappa} \longleftarrow F_{i\kappa}\frac{(XG)_{i\kappa}}{(FG^TG)_{i\kappa}} \tag{14}$$

This alternative interpretation of the Non-negative Matrix Factorisation algorithm allows the classical hard clustering task based on the same cluster selection procedure as in NMF (Eq. 11). Furthermore, it also represents a probabilistic interpretation of the clustering problem supplying degrees of membership of documents to clusters. This information can also be exploited in the recommendation problem as we shall explain in the next section.

4.2. Neighbour Selection Based on PPC

As described before, we want PPC to find better neighbourhoods (clusters) for the users. Therefore, we have to adopt certain decisions in order to model the neighbour selection problem in recommender systems with the PPC algorithm. Which representation fits better this particular problem determines our first decision. In the recommendation problem, the role of documents will be played by users and the role of terms will be played by items which, in collaborative filtering, are the constituent elements of the user representation. In this context, we apply the PPC algorithm under the following settings. Having a collection of n users and m items, let $X = (X_{ab})$ be the items-to-users matrix. The weight of $X_{ab} = X(i_a, u_b)$ will be the rating assigned by the user u_b to the item i_a, i.e., $\text{rat}(u_b, i_a)$. In this initial approach to the problem, we assign zero weight when no rating was produced by the user to the item.

Given this formulation of the clustering scenario, once the minimisation problem formulated in Eq. 12 is solved, the elements of G^* contain the posterior probabilities of the users given the clusters, i.e., $p(u_b|C_l) = G^*_{bl}$. With this information, traditional neighbour selection can be done as in hard-clustering by assigning each user only to the cluster C_k such that $k = \arg\max_z(G^*_{bz})$ where z ranges from 1 to κ.

Therefore, for each user u we obtain a neighbourhood V as the cluster to which the user belongs. Given this situation we can build a recommender which predicts the rating for user u and item i in the following way (Adomavicius and Tuzhilin, 2005):

$$\widetilde{\text{rat}}(u,i) \quad = \quad \frac{\sum_{v \in V} \text{sim}(u,v)\, \text{rat}(v,i)}{\sum_{v \in V} |\text{sim}(u,v)|} \tag{15}$$

where $\widetilde{\text{rat}}(u,i)$ represents a predicted rating (as opposed to an actual rating, denoted as $\text{rat}(u,i)$); besides in this case we estimate $\text{sim}(u,v)$ as

$$\text{sim}(u,v) \propto p(v|V) = G^*_{cl} \tag{16}$$

provided that the index of user v is c (that is, $v = u_c$) and that $V = C_l$ is the cluster assigned to the target user u.

The only remaining decision is to choose the desired number of neighbours (in our case, the number of clusters that we want to obtain with PPC, i.e., κ). We discuss this point in the following section.

5. Experiments and Results

In this section we present three different experiments and discuss the results by comparing the performance of our proposals presented in Sections 3.2 and 4.2 against standard recommendation techniques.

5.1. Evaluation Methodology

In the evaluation of the recommendation methods, we have used two publicly available datasets commonly named as *Movielens 100K* and *Movielens 1M*[2] which are very popular in the evaluation of recommendation methods. Some characteristics about these datasets are shown in Table 1. Note that these datasets are different, in particular, the smaller dataset is not a subset of the larger one (although the movies are similar, there is no relation between the user information of each dataset); to further emphasise this issue, we have incorporated information about the time span each dataset was collected. Furthermore, as we shall see later we have used the smaller dataset to analyse the sensitivity of our approach to different parameters and the larger one to validate the results.

We performed a standard 5-fold cross-validation evaluation using the splits provided with the collections. This is a typical experimental approach in the recommender systems field, where in each split the 80% of the data is retained in order to produce item recommendations which are evaluated with the 20% of the held out data. Note that this cross-validation has solely evaluation purposes and it is independent from the parameter training. We apply ranking oriented

[2]Both are available at http://www.grouplens.org/node/73

Table 1: Statistics about the datasets used in the experiments.

Dataset	#users	#items	#ratings	Sparsity	Recollection Period
Movielens 100K	943	1,682	100,000	6.30%	1997/1998
Movielens 1M	6,040	3,900	1,000,209	4.24%	2000/2003

metrics which have recently started to be widely used to evaluate recommender systems. Besides, it is not possible to apply metrics such as MAE and RMSE to our approaches, since the proposed methods rank items, but do not generate rating predictions.

The methodology used in the evaluation corresponds to the *TestItems* approach described in (Bellogín, Castells, and Cantador, 2011a), where, for each user, a ranking is generated by predicting a score for every item in the test set, only ignoring those items already rated by the user (i.e., in training). We also tested alternative methodologies, such as the one proposed by (Koren, 2008) where a ranking is generated for each item in the test set based on N additional not-relevant items. We observed similar trends to those reported herein with that methodology in preliminary experiments.

Once a ranking has been generated for each user, e.g., with the TestItems methodology, its performance can be measured using, for instance, the *trec_eval* program[3]. In this way, standard IR metrics such as precision, normalised Discounted Cumulative Gain (nDCG) or Mean Reciprocal Rank could be used (Baeza-Yates and Ribeiro-Neto, 2011), where test ratings are used as groundtruth (for nDCG, the explicit rating value is used as relevance grade). In the following we report effectiveness values for precision at 5 (P@5), precision at 50 (P@50) and normalised discounted cumulative gain with cut-offs at 5 and 10 (nDCG@5 and nDCG@10, respectively). Note that, as already acknowledged in (McLaughlin and Herlocker, 2004) and (Wang et al., 2008a), the rated items in the test users represent only a fraction of the items that the user truly liked, and therefore, the measured metrics may underestimate the true metric values.

Regarding the experimental results, we tuned the values of the parameters λ (amount of smoothing of the relevance models) and κ (number of clusters for PPC) involved in the different compared methods by optimising P@5 on the small *Movielens 100K* collection, that is, we perform a 5-fold cross validation evaluation as described above in this dataset and report the best values for each parameter. In the case of one of the baselines, we did not have to perform this tuning process, since the optimal parameter values for the same collection were previously reported in (Wang et al., 2008a), as we shall point out again in the next section. We also report coverage values following the definition given in (Shani and Gunawardana, 2011) of *user space coverage*, that is, the number of users for which the system is able to recommend at least one item. After tuning the parameters λ and κ on *Movielens 100K*, we evaluate the methods

[3]Available at http://trec.nist.gov/trec_eval/

in the larger *Movielens 1M* collection using the same evaluation methodology as before with the optimal parameters obtained for the first dataset. For this reason, sometimes we will refer to the first dataset as the *training collection*, whereas the second would be the *test collection*.

Finally, to analyse the statistical significance of the results, we performed Wilcoxon Signed-Rank Test (Wilcoxon, 1945), where the performance at user level of two methods are compared. In this case the two paired samples are the concatenation of the user-level effectiveness values of the five different folds.

5.2. Baselines

In this work, we have selected a set of representative baselines from the state of the art we reported in Section 2, either in terms of the algorithmic strategy, or in terms of the performance. More specifically, we compared our proposals with a standard User-Based collaborative filtering method *(UB)* (Resnick et al., 1994) where the neighbourhood is selected among the set of 100 most similar users (according to Pearson's correlation). To further put our results in perspective, we also include a state of the art method which does not use any neighbour selection but it is based on Matrix Factorisation through Singular Value Decomposition (SVD) using 50 dimensions *(MF)* (Koren, 2008) and that is generally among the best performing recommendation methods to date (in terms of error metrics).

Moreover, we also tested against other existing proposals based on modelling of the recommendation problem as an Information Retrieval task, where the main differences to our proposals are discussed in Section 2. We test our methods against the user-based formulation of the probabilistic interpretation of the relevance models for log-based CF proposed in (Wang et al., 2006a) *(UIR-User)*, formulated in the Eq. 16 of that paper, that is:

$$p(i|R_u) \propto \sum_{\substack{v \in L_i \\ c(u,v)>0}} \log\left(1 + \frac{(1-\lambda)p_{ml}(v|u,r)}{\lambda p(v|r)}\right) + |L_i| \log \lambda \tag{17}$$

where the sum is over the set of users who have expressed interest for item i ($v \in L_i$) and, at the same time, the number of items rated in common with the target user u ($c(u,v)$) is greater than zero. The maximum likelihood estimator for the user v given the target user u assuming relevance (r) is estimated as follows:

$$p_{ml}(v|u,r) \propto \frac{c(v,u)}{c(u)}$$

And the probability of a user v assuming relevance is estimated by the count of items rated by the user:

$$p(v|r) \propto c(v)$$

Another included baseline is the user-based model presented in (Wang et al., 2008a) *(User-basedRM)*, which allows for introducing ratings in the probability

estimations. More specifically, we use the Eq. 40a from (Wang et al., 2008a) which goes as follows:

$$p(i|R_u) = \widetilde{\text{rat}}(u,i) = \frac{\sum_{v \in L_i} \text{rat}(v,i) e^{-\frac{1-\cos(u,v)}{h_u^2}}}{\sum_{v \in L_i} e^{-\frac{1-\cos(u,v)}{h_u^2}}} \tag{18}$$

where $\cos(u,v)$ is a cosine kernel based similarity measure (Liu, Lu, and Ma, 2004) between the user u and v represented as vectors in the item space, where the missing ratings can be replaced by a constant value of 0 or by the average rating value. As we have discussed in the related work section, this approach requires a prior learning of the value h_u (the kernel bandwidth window parameter) based on an expectation-maximisation process (Wang et al., 2008a). In order to provide a fair comparison, we shall use here the best value reported in (Wang et al., 2008a), which was tuned on the very same collection ($h_u^2 = 0.79$).

5.3. Results

We present now the conducted experimentation, along with the obtained results, in order to validate our contributions and answer the following research questions: (i) Are Relevance-Based Language Models effective as a framework for modelling the recommendation problem? (ii) Is it possible to achieve a better neighbourhood selection by applying probabilistic clustering techniques? And (iii) is it possible to achieve further improvements by the combination of both approaches?

5.3.1. Experiment 1: Relevance-Based Language Models

In this experiment, we assess the validity of our relevance modelling of the recommendation problem. In order to do so, item recommendations are generated using Eq. 4 and 6, and the neighbourhoods are constructed with traditional nearest neighbours approach. Then, we compare the results obtained with these methods against the baselines presented in Section 5.2. The results of the experiments are presented on Table 2, denoting RM1 the results of the RM1 estimation based on the i.i.d. sampling assumption (Eq. 4) and RM2 the results of the RM2 estimation based on the conditional sampling assumption (Eq. 6). Furthermore, we present in Figure 2 and Figure 3 an analysis on the parameter stability of λ (the amount of smoothing in Jelinek-Mercer) in the *Movielens 100K* collection. In all cases, we use the parameter estimation approach described in Section 3.2.3.

The results reported in Table 2, validate our proposal for the relevance modelling of the recommendation process, showing considerable improvement. Both methods achieve a statistically significant advantage against every baseline. The performance enhancement is considerable over every baseline method (between 120% and 200% of improvement in terms of P@5, depending on the dataset). This clearly indicates that the estimates obtained through our relevance modelling of the recommendation problem are more suitable to obtain good effectiveness values. Profile-expansion style recommendation proves to be

Table 2: Summary of the results for each approach, best values for each collection and metric bolded. Statistical significant improvements according to Wilcoxon Test ($p < 0.01$) w.r.t. MF, UB, User-basedRM, UIR-User, RM1, RM2, PPC, PPC+RM1 and PPC+RM2 are superscripted with a, b, c, d, e, f, g, h and i respectively.

Method	*Movielens 100K (training collection)*				
	P@5	nDCG@5	P@50	nDCG@10	Cvg.
MF	0.081^{bcd}	0.076^{bcd}	0.060^{bcd}	0.074^{bcd}	100%
UB	0.026^{cd}	0.020^{cd}	0.057^{cd}	0.029^{cd}	100%
User-basedRM	0.005	0.003	0.054^{d}	0.018^{d}	100%
UIR-User	0.004	0.002	0.002	0.002	100%
RM1	0.240^{abcdfg}	0.221^{abcdfg}	0.141^{abcdfg}	0.214^{abcdfg}	100%
RM2	0.181^{abcdg}	0.161^{abcdg}	0.089^{abcd}	0.153^{abcdg}	100%
PPC	0.135^{abcd}	0.114^{abcd}	0.108^{abcdf}	0.123^{abcd}	95%
PPC+RM1	$0.320^{abcdefg}$	$0.294^{abcdefg}$	$0.162^{abcdefg}$	$0.282^{abcdefg}$	100%
PPC+RM2	$\mathbf{0.327}^{abcdefg}$	$\mathbf{0.297}^{abcdefg}$	$\mathbf{0.168}^{abcdefgh}$	$\mathbf{0.290}^{abcdefg}$	100%

Method	*Movielens 1M (test collection)*				
	P@5	nDCG@5	P@50	nDCG@10	Cvg.
MF	0.062^{bcdg}	0.061^{bcdg}	0.045^{bcd}	0.060^{bcdg}	100%
UB	0.052^{d}	0.049^{d}	0.038^{d}	0.048^{d}	100%
User-basedRM	0.001	0.001	0.034^{d}	0.006^{d}	100%
UIR-User	0.001	0.001	0.001	0.001	100%
RM1	0.205^{abcdfg}	0.192^{abcdfg}	0.112^{abcdfg}	0.182^{abcdfg}	100%
RM2	0.115^{abcdg}	0.109^{abcdg}	0.064^{abcdg}	0.104^{abcdg}	100%
PPC	0.050^{d}	0.044^{d}	0.059^{ad}	0.050^{d}	98%
PPC+RM1	$0.258^{abcdefg}$	$0.243^{abcdefg}$	$0.133^{abcdefg}$	$0.225^{abcdefg}$	100%
PPC+RM2	$\mathbf{0.294}^{abcdefgh}$	$\mathbf{0.275}^{abcdefgh}$	$\mathbf{0.152}^{abcdefgh}$	$\mathbf{0.258}^{abcdefgh}$	100%

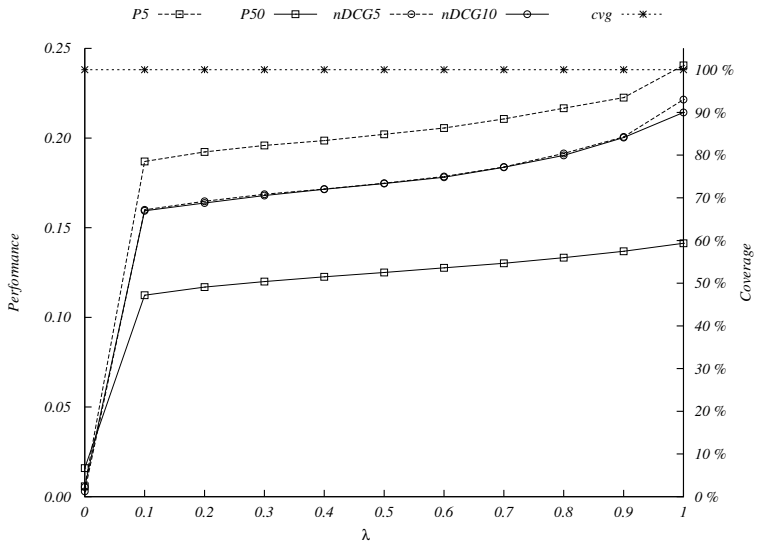

Figure 2: Performance and coverage (cvg) in the Movielens 100K collection when varying the amount of smoothing applied for the RM1 method

a better strategy than pure item ranking based recommendation. The poor behaviour of the UIR-User method was expected because these methods does not exploit rating information but only co-rating. Meanwhile, the User-basedRM, which achieved good results in terms of Mean Absolute Error (MAE) in the original paper, does not perform well in precision oriented tasks. It only achieves comparable results with the other baselines for the $P@50$ metric. The large difference with respect to this method can thus be partly explained by the fact that in the original paper the method is optimised for a different metric from the ones we use here, which are ranking-oriented rather than error-based, as corresponds to a retrieval task.

Overall, this experiment confirms that our proposal of combining neighbourhood information and relevance estimations under the same method is very beneficial to the recommendation task. Furthermore, when analysing the behaviour in terms of the parameter stability in the training collection, it can be observed that both methods are very robust over the parameter values. Meanwhile the optimal λ for the RM1 method is achieved when the amount of applied smoothing is the maximum (in other words, when the background model is used, which just results in a pure popularity-based recommender). This last point is mainly explained by the poor quality of the neighbourhoods created by the nearest neighbour algorithm. This observation is confirmed by the fact that with better neighbour selection algorithms, the best value for λ is no longer the maximum, as we will show in Experiment 3. In the case of the RM2 method, the optimal value is achieved for $\lambda = 0.1$ which indicates that the estimation benefits both from the background model and the users' models. In this case,

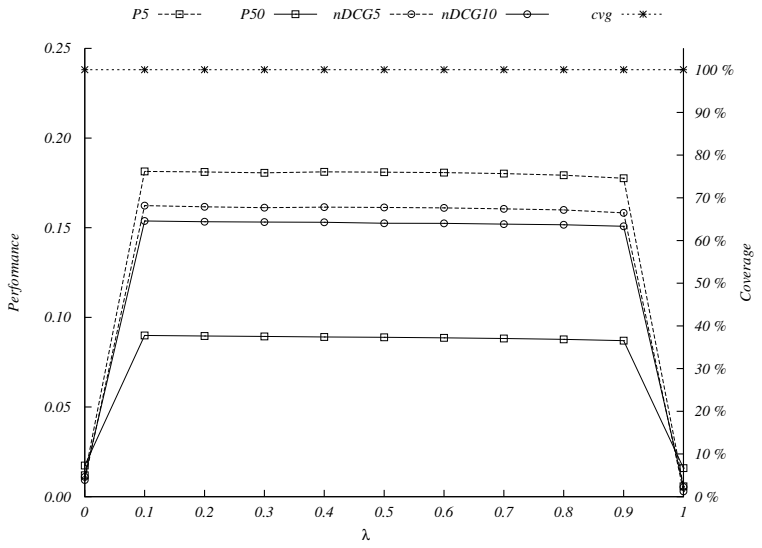

Figure 3: Performance and coverage (cvg) in the Movielens 100K collection when varying the amount of smoothing applied for the RM2 method

the performance of the method based on the conditional sampling assumption is less sensitive than that of the RM1 method.

In this experiment, we used traditional neighbourhood selection techniques for user-based collaborative filtering, that is, based on Pearson's correlation and nearest neighbours. In the next experiment, we assess if more elaborated approaches for such task based on probabilistic clustering can improve the performance of the recommendation process.

5.3.2. Experiment 2: Probabilistic Clustering for Neighbourhood Selection

The objective of this experiment is to evaluate the suitability of the PPC algorithm for the neighbourhood selection task. We followed the experimental set-up described in Section 4.2 and the rating prediction was performed using Eq. 15. The results of applying this method for the neighbourhood selection task instead of using a standard nearest neighbour selection (e.g., computing Pearson's correlation) are presented in Table 2 denoted as PPC. The most important finding is that the neighbourhood selection based on applying probabilistic forms of clustering greatly enhances the performance of the recommendation. Particularly, this method beats every baseline in the training collection, achieving statistically significant improvements.

It is important to highlight, regarding the test collection, that our PPC method outperforms the UB approach for nDCG@10 and every baseline for P@50. We believe the different performance improvements observed for the two collections may be due to the optimal parameter (κ) found in the training collection, which seems to be insufficient for the test collection. This makes sense

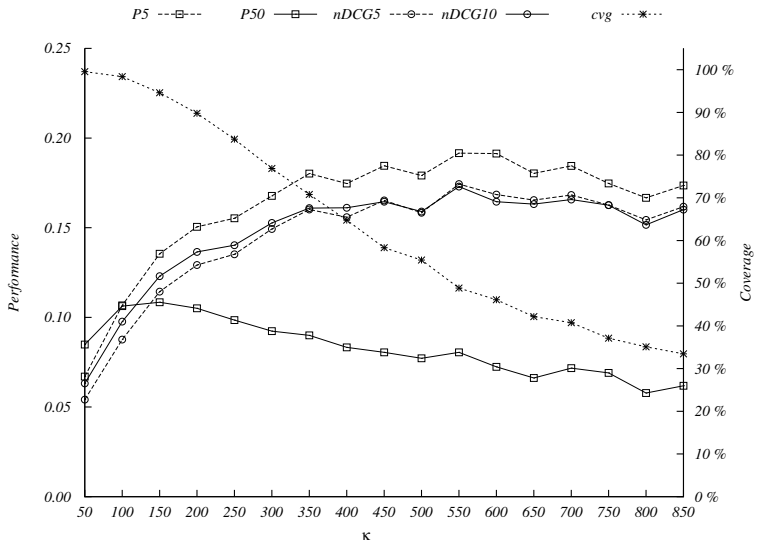

Figure 4: Performance and coverage (cvg) in the Movielens 100K collection when varying the number of clusters PPC method

since the properties of each collection are very different (943 vs 6040 users, see Table 1). The results, nonetheless, are very promising, and underline the fact that improvements of up to 30% for P@50 are possible by tuning on a separate – but not very different – collection and not using the optimal parameters.

As explained before, only one parameter value has to be determined in this experiment, namely the number of clusters κ. In order to study the behaviour of this method when varying the number of clusters, we report the results over the training collection in Figure 4. Interestingly, when increasing the number of clusters the recommendation effectiveness tends to improve but at the expense of coverage. This is explained by the fact that when increasing the number of clusters and, at the same time, working with hard-clustering methods, clusters with very few users tend to appear. For very small clusters, it is not possible to produce a good recommendation for the users belonging to them. It can be observed that a value of $\kappa = 150$ (which corresponds to the values reported for the training collection in Table 2) provides a good trade-off between coverage and effectiveness in this experiment.

5.3.3. Experiment 3: Probabilistic Clustering and Relevance-Based Language Models

Once determined that both approaches, separately, are able to greatly improve the effectiveness of the baselines, we take into consideration the combination of both. In this combination, the neighbourhood selection phase is addressed by applying the PPC method, while the recommendation output is obtained by applying Eq. 4 (PPC+RM1) or Eq. 6 (PPC+RM2). In this case, we have to

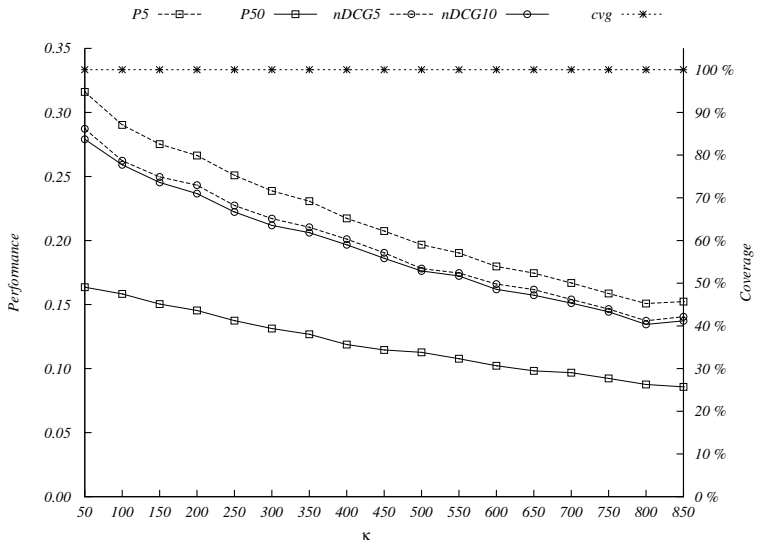

Figure 5: Performance and coverage (cvg) in the Movielens 100K collection when varying the number of clusters PPC+RM1 method

train two parameters, namely, the number of clusters for PPC (κ) and the value of the Jelinek-Mercer smoothing parameter (λ). As in previous experiments, we trained and validated those values in the *Movielens 100K* collection and tested them in the *Movielens 1M* dataset. The results for both collections are summarised in Table 2. The effectiveness of both methods clearly outperforms the four baselines, where these improvements are statistically significant.

Moreover, these combinations also outperform the isolated application of the two approaches – relevance modelling (RM1 and RM2) and probabilistic neighbour selection (PPC) – outperforming the results obtained for Experiments 1 and 2 in every analysed metric. Note that, in this situation, the variation in performance when using the different RM estimations – together with a neighbourhood based on PPC clustering – are negligible in the training collection but significant in the test one. Furthermore, the best value is obtained by the method PPC+RM2, which is slightly better than PPC+RM1, just the opposite to what was found in Experiment 1, for both datasets. Finally, the optimal neighbourhood size found in training was the same for both methods ($\kappa = 50$), and the performance decreases when more clusters are considered (see Figures 5 and 6 for a sensitivity analysis in the training collection). Interestingly, in this experiment the best result is obtained without affecting the coverage, an interesting effect since although a user would be isolated in a singleton neighbourhood by means of the PPC, with the RM modelling it will still benefit from the background knowledge of the collection in the recommendation process.

As an additional checking, we show in Table 3 how these methods are sensitive to the value of the λ parameter in the training collection. It can be

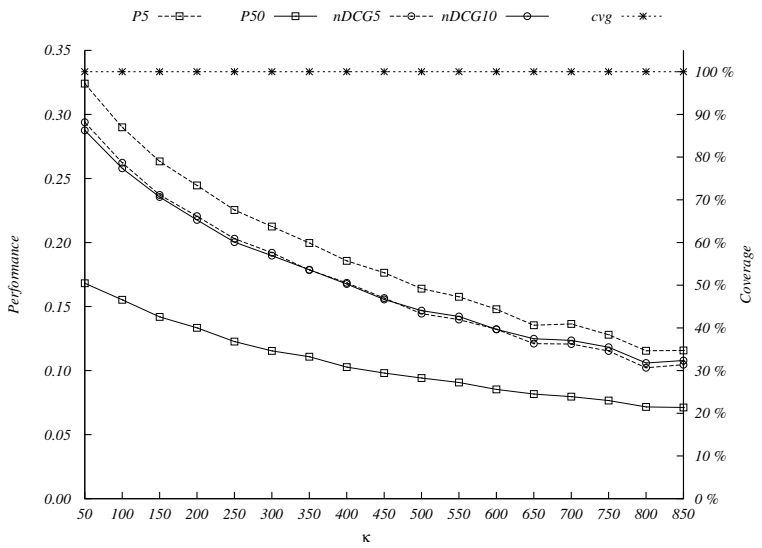

Figure 6: Performance and coverage (cvg) in the Movielens 100K collection when varying the number of clusters PPC+RM2 method

Table 3: Performance results for the combination of PPC and RM models, for P@5 and 50 clusters.

Method	λ value										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
PPC+RM1	0.006	0.316	0.318	**0.320**	0.319	0.317	0.309	0.299	0.288	0.272	0.240
PPC+RM2	0.024	0.324	0.324	0.324	0.325	0.326	0.326	**0.327**	**0.327**	0.326	0.006

observed that we can obtain effectiveness values close to the ones of the best performing λ in a wide range of the parameter space for both methods, stressing the robustness of these approaches.

5.3.4. Discussion

When globally analysing the results of the three experiments, we can conclude that (i) the proposed relevance-based language modelling of the recommendation process performs better than other similar approaches which also capture the relevance notion for this problem, (ii) the probabilistic clustering for neighbourhood selection clearly outperforms traditional neighbourhood selection techniques based on Pearson's correlation on nearest neighbours or matrix factorisation techniques, and (iii) the combination of both approaches enhances even further the recommendation results.

Our relevance model estimations for the recommendation problem have the capability of, depending on the amount of smoothing applied, producing a range of different recommendation strategies, from a pure popularity-based recommendation to a (standard) neighbour-based recommender algorithm. As a result of

this, in the first experiment, we obtained that the best performance of the RM1 method was produced by a popularity-based recommendation, once its optimal configuration of the parameters is analysed. We believe this fact indicates that the quality of the standard neighbourhood techniques is not good enough, and more emphasis on the collection statistics (popularity) should be taken into account. For that reason, we decided to test alternative neighbour selection techniques such as PPC.

From the results of the Experiment 2, we can conclude that PPC obtains better neighbourhoods than standard techniques, in terms of the resulting recommendation performance. In fact, if we compare the results obtained by the classic UB against our method, the precision may be multiplied up to a factor of 5x in the best situation (training collection), whereas a decent improvement of 30% for P@50 has been obtained for the test collection. This improvement is achieved at the expense of lower coverage, but, as shown in Figure 4, even when few clusters are exploited (and, hence, coverage is still high) precision is doubled. Moreover, for the reported values the coverage is quite high, as we may observe in Table 2.

Finally, analysing the results of Experiment 3, we may observe that now the method based on RM1 is not producing solely popularity-driven recommendations for its optimal parameters, but instead a combination of the background model and the neighbourhood information. This is additional evidence supporting the quality of the neighbourhoods obtained by the PPC method. As another consequence, we see in this experiment an important improvement in terms of effectiveness with respect to the results obtained in Experiments 1 and 2.

In summary, the combination of Relevance Modelling and PPC approaches leads to more robust techniques (since the sensitivity of λ has decreased and coverage is now independent from the number κ of clusters), more computationally efficient algorithms (because lower values for κ are required), and better performing techniques in terms of precision and nDCG. Moreover, since the optimal parameters found in the training collection have proved to be effective in the test collection, we may conclude that our methods are also flexible and general enough to be trained and tested in two collections with different properties while showing good performance in both situations.

6. Conclusions and Future Work

In this paper, we have proposed a relevance modelling approach for the recommendation problem. Our proposal addresses the item recommendation task as a profile expansion problem, using the mechanisms for query expansion provided by the Relevance-Based Language Models. The adaptation of this IR model to the recommendation task is non-trivial, given the different nature – and even the number – of the input data spaces the systems handle in each case. We devise an adaptation where the recommendation variables (items and users) play different roles at different points in the model, thus resulting in a recommendation RM where the elements have a quite different meaning than

intended in the original formulation for text IR. The development of the model into terms that link to available observations is quite different as well. Yet we still inherit the practical and theoretical advantages of language models in IR, such as the avoidance of an explicit relevance variable, which makes the resulting probabilistic framework easier to develop and bring to computable statistical estimations (e.g. no need for explicit relevance judgments in the recommendation model), while retaining a clear formal underpinning.

The empirical evaluation of our proposal shows significant improvements in terms of effectiveness (measured by ranking quality metrics) against different related baselines. Furthermore, in order to obtain better neighbours for the memory-based recommendation, we proposed the application of the Posterior Probabilistic Clustering algorithm. This proposal by itself also achieves effectiveness improvements over traditional neighbour-based approaches, while at the same time it outperforms standard matrix factorisation algorithms and other probabilistic-based approaches. Furthermore, we show that the combination of both proposals improves the results of their individual application, demonstrating in this way that the better the neighbourhood (which acts as the pseudo relevance set in the explicit search scenario), the better the estimations of the underlying relevance model, and therefore, better item recommendations are produced as expansions of the user profile. This fact is consistent with previous results obtained in the application of RM on text retrieval.

Several potential directions open up from this point to improve the recommendation effectiveness further. We plan to further study other options for the construction of the pseudo relevance set of users, not only techniques based on neighbours but also other approaches to produce an initial user ranking, as is standard in text retrieval. We will also consider alternative estimations and smoothing approaches to be applied in our formulation of the problem. We envision additional refinements of our methods, such as only considering positively rated items in the user profile when computing the user likelihood, or tackling differently the absence of rating for an item by the user in the PPC algorithm. We also plan to explore the use of our approach as a basis to address the problem of recommendation diversification. We envision the diversification of the recommended items in the expanded profile, as an equivalent problem to promoting divergent terms in the estimation of the relevance models (Parapar and Barreiro, 2011). Finally, we aim to study the formulation of the equivalent item-based recommendation modelling corresponding to our user-based proposal. This technique is known to perform better than the user-based in some situations, and thus, the relevance model approach might find performance improvements also in those scenarios.

Vitae

Javier Parapar (`http://www.dc.fi.udc.es/~parapar/`) is a Ph.D. student in the IRLab, Department of Computer Science, University of A Coruña. He holds a MS.c. in Computer Science from the same University. His research interests comprise: pseudo-relevance feedback, clustering and

cluster based retrieval, blog and news search, document processing and engineering, text summarisation and retrieval over degraded information.

Alejandro Bellogín (`http://ir.ii.uam.es/~alejandro/`) is a Ph.D. student in the Information Retrieval Group at the Computer Science Department, Autónoma University of Madrid. He holds a M.Sc. in Computer Science and Telecommunications from the same University. His research is focused on recommender systems, in particular, adaptations from the information retrieval area, such as performance prediction techniques, evaluation methodologies, and probabilistic models.

Pablo Castells (`http://ir.ii.uam.es/castells/`) holds a Ph.D. from the Autónoma University of Madrid (UAM). He leads the Information Retrieval Group (`http://ir.ii.uam.es`) in the Computer Science Department at UAM, where he is an Associate Professor. His research interests focus on IR models, evaluation, personalisation, recommender systems, and IR diversity. He has coordinated several research projects funded by European and national programmes.

Álvaro Barreiro (`http://www.dc.fi.udc.es/~barreiro/`) holds a Ph.D. from University of Santiago de Compostela. He is a Professor at the Department of Computer Science of the University of A Coruña where he leads the Information Retrieval Lab (`http://www.irlab.org`). He has been the main researcher of several IR research projects funded by the Spanish Government.

References

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

Baeza-Yates, R. A., Ribeiro-Neto, B. A., 2011. Modern Information Retrieval - the concepts and technology behind search, Second edition. Pearson Education Ltd., Harlow, England.

Balasubramanian, N., Allan, J., Croft, W. B., 2007. A comparison of sentence retrieval techniques. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR '07. ACM, New York, NY, USA, pp. 813–814.

Bellogín, A., Castells, P., Cantador, I., 2011a. Precision-oriented evaluation of recommender systems: an algorithmic comparison. In: Proceedings of 5th ACM Conference on Recommender Systems. RecSys'11. pp. 333–336.

Bellogín, A., Parapar, J., 2012. Using graph partitioning techniques for neighbour selection in user-based collaborative filtering. In: RecSys '12: Proceedings of the sixth ACM conference on Recommender systems. ACM, New York, NY, USA, pp. 213–216.

Bellogín, A., Wang, J., Castells, P., 2011b. Structured collaborative filtering. In: Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011. pp. 2257–2260.

Bellogín, A., Wang, J., Castells, P., 2011c. Text retrieval methods for item ranking in collaborative filtering. In: Advances in Information Retrieval - 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings. pp. 301–306.

Billsus, D., Pazzani, M. J., 1998. Learning collaborative information filters. In: Shavlik, J. W. (Ed.), ICML. Morgan Kaufmann, pp. 46–54.

Collins-Thompson, K., Callan, J., 2007. Estimation and use of uncertainty in pseudo-relevance feedback. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR'07. ACM, New York, NY, USA, pp. 303–310.

Desrosiers, C., Karypis, G., 2011. A comprehensive survey of neighborhood-based recommendation methods. In: Recommender Systems Handbook. pp. 107–144.

Ding, C., Li, T., Luo, D., Peng, W., 2008. Posterior probabilistic clustering using nmf. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR '08. ACM, New York, NY, USA, pp. 831–832.

Gu, Q., Zhou, J., Ding, C. H. Q., 2010. Collaborative filtering: Weighted non-negative matrix factorization incorporating user and item graphs. In: Proceedings of the SIAM Conference on Data Mining. SDM 2010. pp. 199–210.

Herlocker, J. L., Konstan, J. A., Riedl, J., 2002. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Information Retrieval 5 (4), 287–310.

Hill, W., Stead, L., Rosenstein, M., Furnas, G., 1995. Recommending and evaluating choices in a virtual community of use. In: Proceedings of the SIGCHI conference on Human factors in computing systems. CHI '95. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp. 194–201.

Koren, Y., 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. KDD '08. ACM, pp. 426–434.

Lafferty, J., Zhai, C., 2002. Probabilistic relevance models based on document and query generation. In: Language Modeling and Information Retrieval. Kluwer Academic Publishers, pp. 1–10.

Lavrenko, V., Croft, W. B., 2001. Relevance-based language models. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR'01. pp. 120–127.

Lee, D. D., Seung, H. S., Apr. 2001. Algorithms for Non-negative Matrix Factorization. In: Leen, T. K., Dietterich, T. G., Tresp, V. (Eds.), Advances in Neural Information Processing Systems 13. MIT Press, pp. 556–562.

Lee, K. S., Croft, W. B., Allan, J., 2008. A cluster-based resampling method for pseudo-relevance feedback. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR'08. ACM, New York, NY, USA, pp. 235–242.

Li, X., Zhu, Z., 2008. Enhancing relevance models with adaptive passage retrieval. In: Proceedings of the 30th European conference on Information Retrieval. ECIR'08. Springer-Verlag, Berlin, Heidelberg, pp. 463–471.

Liu, Q., Lu, H., Ma, S., 2004. Improving kernel Fisher discriminant analysis for face recognition. Circuits and Systems for Video Technology, IEEE Transactions on 14 (1), 42–49.

Lv, Y., Zhai, C., 2009. A comparative study of methods for estimating query language models with pseudo feedback. In: Proceeding of the 18th ACM conference on information and knowledge management. CIKM '09. ACM, New York, NY, USA, pp. 1895–1898.

McLaughlin, M. R., Herlocker, J. L., 2004. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR '04. ACM, New York, NY, USA, pp. 329–336.

O'Connor, M., Herlocker, J., 1999. Clustering items for collaborative filtering. In: ACM SIGIR Workshop on Recommender Systems.

Parapar, J., Barreiro, A., 2011. Promoting divergent terms in the estimation of relevance models. In: Proceedings of the Third international conference on Advances in information retrieval theory. ICTIR'11. Springer-Verlag, Berlin, Heidelberg, pp. 77–88.

Parapar, J., Barreiro, A., 2012. Language modelling of constraints for text clustering. In: Proceedings of the 34th European Conference on Information Retrieval. ECIR'12. Springer-Verlag, Berlin, Heidelberg, pp. 352–363.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J., 1994. Grouplens: An open architecture for collaborative filtering of netnews. In: CSCW. pp. 175–186.

Shani, G., Gunawardana, A., 2011. Evaluating recommendation systems. In: Recommender Systems Handbook. pp. 257–297.

Wang, J., 2009. Language models of collaborative filtering. In: Proceedings of the 5th Asia Information Retrieval Symposium on Information Retrieval Technology. AIRS '09. Springer-Verlag, Berlin, Heidelberg, pp. 218–229.

Wang, J., de Vries, A., Reinders, M., 2006a. A user-item relevance model for log-based collaborative filtering. In: Proceedings of the 28th European Conference on Information Retrieval. ECIR'06. Springer-Verlag, Berlin, Heidelberg, pp. 37–48.

Wang, J., de Vries, A. P., Reinders, M. J. T., 2006b. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR'06. pp. 501–508.

Wang, J., de Vries, A. P., Reinders, M. J. T., Jun. 2008a. Unified relevance models for rating prediction in collaborative filtering. ACM Trans. Inf. Syst. 26 (3), 16:1–16:42.

Wang, J., Robertson, S., de Vries, A. P., Reinders, M. J., Dec. 2008b. Probabilistic relevance ranking for collaborative filtering. Information Retrieval 11 (6), 477–497.

Wilcoxon, F., 1945. Individual Comparisons by Ranking Methods. Biometrics Bulletin 1 (6), 80–83.

Xu, W., Liu, X., Gong, Y., 2003. Document clustering based on non-negative matrix factorization. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. SIGIR '03. ACM, New York, NY, USA, pp. 267–273.

Xue, G. R., Lin, C., Yang, Q., Xi, W., Zeng, H. J., Yu, Y., Chen, Z., 2005. Scalable collaborative filtering using cluster-based smoothing. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. SIGIR'05. ACM, pp. 114–121.

Zhai, C., Lafferty, J., 2004. A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. 22 (2), 179–214.

Zhang, S., Wang, W., Ford, J., Makedon, F., 2006. Learning from incomplete ratings using non-negative matrix factorization. In: Proceedings of the SIAM Conference on Data Mining. SDM'06.