ABSTRACT

In the Recommender Systems field ensemble techniques gain growing interest. This approach is based on the idea of mixing many recommenders and to get an average prediction from all of them. Even if it is useful this process may be very expensive from a computational point of view. We propose the use of Operations Research techniques in order to optimize the balance of different predictors and to accelerate it. We show that this problem can be generalized, thus we provide a mathematical framework which helps to find further improvements.

Categories and Subject Descriptors
H.3.3 [Information Search and Retrieval]: Information filtering; H.3.4 [Systems and Software]: Performance evaluation.; G.1.6 [Optimization]: Nonlinear programming

General Terms
Algorithms, Measurement, Experimentation.

Keywords
Recommender Systems, Collaborative Filtering, Optimization.

1. INTRODUCTION

Recommender Systems represent already a successful technology and have a strong foundation [6, 12, 11]. By the way, many research groups are proposing possible extensions in order to overcome some limitations [1]. Recent trends suggest the use of ensemble techniques in order to improve the quality of recommendations. This approach is based on the idea of mixing many recommenders and to get an average prediction from all of them. Instead of using only one predictor, many are used together and they cooperate to produce the ultimate recommendation which is some kind of combination (blend) of the ones proposed by each of them.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

2. THE BLENDING PROBLEM

In order to formally define our problem in OR style we have to identify these key elements:

• objective function;
• decision variables;
• constraints.

Typically, the objective function has to be minimized and represents the task, decision variables have to be chosen in a feasible region and represent choices and constraints determine the feasible area and represent limits.

We begin by introducing a very general formalization (basically a schema) and refine it step by step.

2.1 Problem Schema

Let $M$ be a set of predictors, $P_{Pr_m}$ an element of this set, $\beta(M)$ a blending operator which mixes different predictors, $\lambda$ a vector of design variables, $P_{\nu}(\lambda)$ constraints that $\lambda$ must respect and $E$ an error function which measures the quality of recommendations.

Thus, the schema of the Blending Problem is:
min \lambda \mathcal{E}(\beta(M,\lambda)) \\
\text{s.t.} \ P_n(\lambda)
(1)

2.2 Error Function
First of all we have to define the error function \( \mathcal{E} \) which expresses the error done by the system. Many different measures are used in order to evaluate the performance of filtering algorithms employed by Recommender Systems and some metrics fit better for top-N recommendation, and others for prediction. The glamour of the Netflix competition has made one metric notorious: accuracy computed as the square root of the averaged squared difference between each prediction and the actual rating (the root mean squared error or “RMSE”). We believe also that it represents a good choice (see [9] for a discussion on this topic).

Let the \( r_{ui} \) denote the actual rating provided by a certain user \( u \) for an item \( i \), with \( i = 1, 2, \ldots, n_u \) (\( n_u \leq n \), where \( n \) is the number of all available items) and let \( p_{ui} \) denote the prediction generated by a certain algorithm for the same user and the same item. The RMSE for a user becomes:

\[
RMSE_u = \sqrt{\frac{\sum_{i=1}^{n_u} (r_{ui} - p_{ui})^2}{n_u}}
(2)
\]

The total RMSE can be obtained as an average of the RMSE of all users:

\[
RMSE = \sqrt{\frac{\sum_u \sum_{i=1}^{n_u} (r_{ui} - p_{ui})^2}{\sum_u n_u}}
(3)
\]

2.3 Blending Function
In order to mix different predictors we may simply calculate their linear combination. Thus, we adopt the weighted average as a specialization of the blending operator \( \beta \). Let \( p_{mui} \) be prediction generated by predictor \( P_m \) for the user \( u \) and the item \( i \). The RMSE for a user becomes:

\[
RMSE_u = \sqrt{\frac{\sum_{i=1}^{n_u} (r_{ui} - \frac{\sum_{m=1}^{M} w_{mu} p_{mui}}{\sum_{m=1}^{M} w_{mu}})^2}{n_u}}
(4)
\]

We look for the vector of weights \([w_{1u}, \ldots, w_{Mu}]\) which would have minimized the error. An important remark is that the weights are the decision variables and (4) is the objective function; thus ratings and predictions are known parameters. Moreover, we underline that we need a different vector of weights for each user. Basically, we have to solve as many specific optimization problems as the total number of users; in the reminder we call it personalized blending.

On the other hand, a different approach is the global blending. We look for a single vector of weights, that is independent from the users. In this case, the RMSE is:

\[
RMSE = \sqrt{\frac{\sum_u \sum_{i=1}^{n_u} (r_{ui} - \frac{\sum_{m=1}^{M} w_{mu} p_{mui}}{\sum_{m=1}^{M} w_{mu}})^2}{\sum_u n_u}}
(5)
\]

The evaluation of RMSE is typically performed using the “leave-n-out” approach [4], where a part of the dataset is hidden and the rest is used as a training set for the recommender, which tries to predict properly the withheld ratings.

2.4 Reformulations
The objective (4) is a non linear form which is not easy to optimize, so we look for another way to formulate the problem. A first observation is that we can simplify the equations of RMSE ignoring the square root (because we consider only positive values). Secondly, we can introduce a constraint on the sum of weights, so for the personalized blending we get only one optimization problem:

\[
\begin{align*}
\min_{w} & \quad \sum_{i=1}^{n_u} (r_{ui} - \frac{\sum_{m=1}^{M} w_{mu} p_{mui}}{\sum_{m=1}^{M} w_{mu}})^2 \\
\text{s.t.} & \quad \sum_{m=1}^{M} w_{mu} = 1
\end{align*}
(6)
\]

We recall that we should solve one optimization problem like (6) for each user. On the other hand, for the global blending we get:

\[
\begin{align*}
\min_{w} & \quad \sum_{i=1}^{n_u} (r_{ui} - \frac{\sum_{m=1}^{M} w_{mu} p_{mui}}{\sum_{m=1}^{M} w_{mu}})^2 \\
\text{s.t.} & \quad \sum_{m=1}^{M} w_{m} = 1
\end{align*}
(7)
\]

We plan to use these formulations to perform initial computational tests. Anyway, if we want be general we may relax the constraint, and recognise that we got a sum of squares linear fractional problem whose general mathematical programming problem (SSLFP) can be formulated as follows:

\[
\begin{align*}
\min & \quad \sum_{i=1}^{n} \left( \frac{\sum_{j=1}^{n} a_{ij} x_j + b - p_i}{d^2 + e} \right)^2 \\
\text{s.t.} & \quad x_i \geq 0 \quad \forall i \leq n
\end{align*}
(8)
\]

Where \( x \in \mathbb{R}^n, a, d \in \mathbb{R}^\ast, p \in \mathbb{R}^n \). This formulation is a general, hence very useful in order to look for good reformulations. “It is well known that several different formulations may share the same numerical properties (feasible region, optima) though some of them are easier to solve than others with respect to the most efficient available algorithms . . . When a problem with a given formulation \( P \) is cast into a different formulation \( Q \), we say that \( Q \) is a reformulation of \( P \).” [8].

2.5 Conclusion and future work
We plan to move towards two different objectives. Firstly we will perform computational tests based on (6) and (7). This activity is planned as follows:

- identification of the set of predictors, including well-known ones as KNN [5] and Slope-One [7];
- choice of the dataset, for example the ones provided by GroupLens (http://www.grouplens.org);
- selection of the solvers for the problems (6) and (7); since they are non-linear problems, we can use well-known non-linear solvers like COUENNE [3] or BARON [10];
- comparison of the results obtained with (6) and (7); in our opinion, (6) allows to obtain a better level of accuracy with respect to (7), while the time complexity of (6) is greater than the complexity of (7) by a factor that is the total number of user.

Secondly, we want to investigate the best way to reformulate the problem starting from (8).
3. REFERENCES


