Novel method for users' profiles construction through collaborative filtering

Oumayma BANOUAR†, Said RAGHAY†

†Faculty of Science and Techniques, University Cadi Ayyad, Marrakesh, Morocco

Summary
Current information systems provide transparent access to multiple, distributed, autonomous and potentially redundant data sources. Their users may not know the sources they questioned, nor their description and content. Consequently, their queries reflect no more a need that must be satisfied but an intention that must be refined. The purpose of the personalization is to facilitate the expression of users' needs. It allows them to obtain relevant information by maximizing the exploitation of their preferences grouped in their respective profiles.
In this work, we present a matrix completion approach that minimize the nuclear norm to construct users' profiles. The initial data matrix corresponds to the ratings provided by users to items. Each row or column contains at least one observation. The proposed approach, based on collaborative filtering concept, starts by a learning process to classify users and preferences. It exploits then these clusters to run a predictive method in the aim to recover the missing or unknown data. Finally, it uses an assignment function to find the ratings of preferences that were not included in the initial data matrix due to the fact of lack of observation.

Key words:
Personalization, enrichment, user query, user profile, collaborative filtering, bi-clustering, matrix completion, aggregation, assignment function.

1. Introduction

The multiplicity of data sources, their scalability and the increasing difficulty to control their descriptions and their contents are the reasons behind the emergence of the need of users' requests personalization. A major limitation of these systems is their inability to classify and discriminate users based on their interests, their preferences and their query contexts. They cannot deliver relevant results according to their respective profiles[1]. Consequently, the execution of the same request expressed by different users over an ontology-based mediation system will necessarily not provide the same results.

We talk here about a personalized access to data sources using ontologies. A user accessing an information system with the intention of satisfying an information need, may have to reformulate the query issued several times and sift through many results until a satisfactory, if any, answer is obtained. This is a very common experience. A critical observation is that: different users may find different things relevant when searching, because of different preferences, goals etc. Thus, they may expect different answers to the same query. The personalization of a query uses the user profile to rephrase his request by integrating elements of his interests or his preferences. Storing user preferences in a user profile gives a retrieval system the opportunity to return more focused, personalized and hopefully smaller answers.

The objective of the query personalization process is to enhance the user query with his related preferences stored in his profile. The step of user profile construction is the key enabler of an efficient enrichment or personalization process. It consists of predicting missing preferences from a sampling of observations about users. It must focus on the system user and must enable the exploitation of what is called personal relevancy [2] instead of consensus relevancy. In the first one, the information system computes relevancy based on each individual's characteristics, unlike the second one where it presumes that the relevancy computed for the entire population is relevant for each user that is the case for the existing matrix completion methods via convex programming. These methods work on the assumption that at least one observation exists per row and per column.

This work presents a matrix completion method based on the optimisation of the nuclear norm of the matrix that represents the preferences of our system users over items. It uses a bi-clustering process to detect users respectively items clusters in the aim to promote the personal relevancy concept by applying a matrix completion process over users that chairs almost the same preferences. It exposes a more complete method for user profile construction that allows, in addition, the possibility to predict ratings of unrated preferences in the initial users-ratings matrix.

The remaining of this paper is organised as follows. Section 2 discusses the problematic of matrix completion and the possibility of resolving it by using convex programming. Section 3 presents the proposed approach where in section 4 numerical results are shown. Finally, we conclude by exposing the next challenges for data management using learning methods in an information retrieval context.

Manuscript received August 5, 2017
Manuscript revised August 20, 2017

2. Matrix completion

In many practical problems of interest, one would like to guess the missing entries of an $n_1 \times n_2$ matrix from a sampling Ω of its entries. This problem is known as the matrix completion problem. It comes up in a great number of applications including those of collaborative filtering. The collaborative filtering is the task of making automatic predictions about the interests of a user by collecting taste information from many users. In each instance, the objective is to predict the preferences of a user for all items from:

- A partial list of his preferences for a few rated items.
- Information gleaned from many others users.

In mathematical terms, this problem is posed as follows:

A data matrix $M \in \mathbb{R}^{n_1 \times n_2}$ is the matrix to be known as much as possible. The only information available about it is a sampling set of entries $M_{ij}, (i,j) \in \Omega$, where Ω is a subset of the complete set of entries $\{1, \dots, n_1\} \times \{1, \dots, n_2\}$.

Very few factors contribute to an individual's taste. Then, the problem of matrix completion is a problem of a low-rank r matrix from a sample of its entries. The matrix rank satisfies $r \leq \min(n_1, n_2)$.

Such a matrix is represented by counting $n_1 \times n_2$ numbers but has only $r \times (n_1 + n_2 - r)$ degrees of freedom. When the matrix rank is small and its dimension is large, then the data matrix carries much less information than its ambient dimension suggests.

Users, rows of the matrix, are given the opportunity to rate items, columns of the data matrix. However, they usually rate very few ones so there are very few scattered observed entries of this data matrix. In this case, the users-ratings matrix is approximately low-rank because as mentioned, it is commonly believed that only very few factors contribute to an individual's tastes or preferences. These preferences are stored in a user profile.

Let $P_\Omega: \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^{n_1 \times n_2}$ be the orthogonal projection onto the subspace of matrices that vanish outside of Ω. $(i,j) \in \Omega$ if and only if M_{ij} is observed. $Y = P_\Omega(X)$ is defined as:

$$Y_{ij} = \begin{cases} X_{ij}, & (i,j) \in \Omega, \\ 0, & otherwise, \end{cases} \quad (1)$$

The data known in M is given by $P_\Omega(M)$. The matrix M is recovered then from $P_\Omega(X)$ if it is the unique matrix of rank less or equal to r and consistent with the data, which means that, M is the unique solution to:

$$\begin{aligned} &minimize\ rank(X) \\ &subject\ to\ P_\Omega(X) = P_\Omega(M) \end{aligned} \quad (2)$$

In practical point of view, the rank minimization problem is an NP-hard problem. Algorithms are not capable to resolve it in time once the matrices have an important dimension. They require time doubly exponential in the dimension of the matrix to find the exact solution.

2.1 Nuclear norm and matrix model

Authors in [3] proposed the resolution of matrix completion problem by solving the nuclear norm minimization problem:

$$\begin{aligned} &minimize\ \| X \|_* \\ &subject\ to\ P_\Omega(X) = P_\Omega(M) \end{aligned} \quad (3)$$

Where the nuclear norm $\| X \|_*$ is defined as the sum of its singular values:

$$\| X \|_* := \sum_i \sigma_i(X). \quad (4)$$

Matrix completion problem is not as ill posed as thought. It is possible to resolve it by convex programming. The rank function counts the number of nonvanishing singular values when the nuclear norm sums their amplitude. The nuclear norm is a convex function. It can be optimized efficiently via semidefinite programming.

Theorem 1: Let M be an $n_1 \times n_2$ matrix of rank r sampled from the random orthogonal model, and put $n = \max(n_1, n_2)$. Suppose we observe m entries of M with locations sampled uniformly at random. Then they are numerical constants C and c such that if

$$m \geq Cn^{\frac{5}{4}} r log n \quad (5)$$

The minimizer to the problem (3) is unique and equal to M with probability at least $1 - cn^{-3}$; that is to say, the semidefinite program (3) recovers all the entries of M with no error. In addition, if $r \leq n^{1/5}$, then the recovery is exact with probability at least $1 - cn^{-3}$ provided that

$$m \geq Cn^{6/5} r log n \quad (6)$$

Under the hypothesis of Theorem 1, there is a unique low-rank matrix, which is consistent with the observed entries. This matrix can be recovered by the convex optimization (3). For most problems, the nuclear norm relaxation is formally equivalent to the combinatorial hard rank minimization problem.

If the coherence is low, few samples are required to recover M. As an example of matrices with incoherent column and row space matrices with random orthogonal model or those with small components of the singular vectors of M.

2.2 Related works

Conventional semidifinite programming solvers such as SDPT3 [4] and SeDeMi [5] solves the problem (3). However, such solvers are usually based on interior-point methods, and can not deal with large matrices. They can only solve problems of size at most hundreds by hundreds on a moderate PC. Therefore, the first-order methods are used to complete large low rank matrices by solving (3).

The singular value thresholding SVT algorithm approximates the minimization (3) by:

$$min_X\ \tau \parallel X \parallel_* + \frac{1}{2} \parallel X \parallel_F{}^2 \quad s.t.\ \mathrm{X}_{ij} = \mathrm{M}_{ij}, (i,j) \in \Omega \tag{7}$$

With a large parameter τ . $\|.\|_F$ denotes the matrix Frobenius norm or the square root of the summation of squares of all entries.

Then, it applies a gradient ascent algorithm to its dual problem. The iteration is:

$$\begin{cases} X_k = \mathcal{D}_\tau(Y_{k-1}), \\ Y_k = Y_{k-1} + \delta_k P_\Omega(M - X_k), \end{cases} \tag{8}$$

Where $\mathcal{D}_\tau$ is the SVT operator defined as:

$$\mathcal{D}_\tau(Y) \coloneqq arg\ min_{\ X \in \mathbb{R}^{n_1 \times n_2}}\ \frac{1}{2} \parallel Y - X \parallel_F + \tau \parallel X \parallel_*, \tag{9}$$

The iteration is called the SVT algorithm and it was shown to be an efficient algorithm for huge low rank matrices.

Authors in [6] presented the FPCA algorithm. It combines the fixed-point continuation [7] with Bregman iteration [8]. The iteration is as follows:

$$\begin{cases} Iterate\ on\ i\ to\ get\ X_k \begin{cases} X_i = \mathcal{D}_\tau(Y_{i-1}), \\ Y_i = X_{i-1} + \delta_i P_\Omega(M + Z_{k-1} - X_i), \end{cases} \\ Z_k = Z_{k-1} + P_\Omega(M - X_k) \end{cases} \tag{10}$$

In fact, the FPCA algorithm is a gradient ascent algorithm applied to an augmented Langrangian of (3).

The augmented Langrangian Multiplier method ALM in [9], reformulates the problem into

$$\min_{\mathrm{X}} ||\mathrm{X}||_* \quad \text{s.t. } \mathrm{X} + \mathrm{E} = \mathrm{P}_\Omega(\mathrm{M}), \quad \mathrm{P}_\Omega(\mathrm{E}) = 0, \tag{11}$$

Where E is an auxiliary variable. The corresponding (partial) ALM function is

$$\mathcal{L}(X,E,Y,\mu) = \big|\big|X\big|\big|_* + \langle Y, P_\Omega(M) - X - E\rangle + \frac{\mu}{2}\big|\big|P_\Omega(M) - X - E\big|\big|_F^2, \tag{13}$$

$with\ \mathrm{P}_\Omega(E) = 0.$

An inexact gradient ascent is applied to the ALM and leads to the following algorithm:

$$\begin{cases} X_k = \mathcal{D}_{\mu_k^{-1}}(P_\Omega(M) - E_{k-1} + \mu_k^{-1} Y_{k-1}), \\ E_k = -P_{\Omega^c}(X_k), \\ Y_k = Y_{k-1} + \mu_k P_\Omega(M - X_k). \end{cases} \tag{14}$$

For all these algorithms, the SVT operator is the key to make them converge to low rank matrices.

The objective of this paper is to construct users' profiles by applying a matrix completion process. Indeed, for each user, the problem corresponds to predict the unknown ratings to preferences from the observed ones or based on other users preferences. We aim to focus on the system users and to enable what is called personal relevancy instead of consensus relevancy.

In the first one, the information system computes relevancy based on each individual's characteristics, unlike the second one where it presumes that the relevancy computed for the entire population is relevant for each user, which is the case for the algorithms presented above. To have any hope of recovering an unknown matrix using these algorithms, one needs at least one observation per row and one observation per column.

3. Proposed approach

The users' profiles construction is the key personalization enabler and the useful tactic in data integration tasks dealing with irrelevancy problem. It takes elements of the user preferences as input and determines his profile as output. A user profile is a set of weighted elements that defines preferences of its owner over items. Machine learning approach enable the possibility to manipulate profiles automatically as much as possible.

Our idea here is to promote the concept of personal relevancy.

The proposed approach then is based on three main steps:

- A learning process to identify users and preferences clusters.
- A predictive method using clusters found in step 1 to predict the unknown data.
- An assignment function to find the ratings of unrated preferences.

The first step of our approach is to perform a data filtering. The users-ratings matrix R contains only the preferences rated by at minimum one user. The learning process then

starts by applying a Principal Component Analysis PCA in the attempt to reduce the the number of variables and make the information less redundant. As a result, our data are centered. They simulate the contributions of users over preferences.

To detect the users respectively preferences clusters, the process uses a bi-clustering step by using the K-means algorithm on the principal component scores; that is, the representation of the data matrix in the principal component space and its correlation matrix. The second process takes place to predict the missing ratings.

For a given user, respectively an item, we identify clusters in which the selected user, respectively the preference, belongs. The predicted score or rate is the result of Singular Value Thresholding SVT algorithm [3] applied on the matrix containing rates that users in the selected user cluster given to preferences in the selected preference cluster .

The adopted algorithm takes as parameters three mandatory elements.

- Ω the set of locations corresponding to the observed entries. It might be defined in three forms. The first one as a sparse matrix where only the elements different of 0 are to take into account. The second one as a linear vector that contains the position of the observed elements. And the third one where Ω is specified as indices (i, j) with $(i, j) \in \mathbb{N}$.
- b the linear vector which contains the observed elements.
- m_u the smoothing degree.

The application of the SVT algorithm in blocks procures in some cases certain results that are out of range. In this case, the process uses an aggregation function to predict the following rates. It is equal to the mode of all rates found by the intersection between the cluster to which the user belongs and the cluster that contains the selected preference.

Algorithm 1: **Predicting the rate of an item given by a user**

Require : R, Pu, Pi, uk, ik'
- R users-rating matrix
- Pu users partitions
- Pi items partitions
- uk user
- ik' item
- mu the smoothing degree

Ensure: $\widehat{R}_{u_k, i_{k'}}$

S ← ∅ { set of rates}
Cu ← {Cu ϵ Pu : uk ϵ Cu}
Ci ← {Ci ϵ Pi : ik' ϵ Ci}
m=size(Cu) ;
n=size(Ci) ;
M(m,n) ; $\widehat{M}(m, n)$;
b← ∅ {set of observed rates}
omega← ∅ {set of indices of observed rates}
for u from 1 to m do
  for i from 1 to n do
    if (exist(R[u,i])
      b.append(R[u,i]);
      omega.append((u,i));
      S.append(R[u,i]);
      M[u,i]=R[u,i];
    end if
  end for
end for
$\widehat{R} = \mathrm{SVT}([m, n], \mathrm{omega}, b, m_u)$;
if (not exist($\widehat{R}_{u_k, i_{k'}}$))
$\widehat{R}_{u_k, i_{k'}} = \mathrm{mode}(S)$;
end if

The result of this algorithm is a completed data matrix that contains all the ratings of users to preferences. However, these preferences correspond only to the ones that were rated by a user at minimum. The proposed prediction process runs on the assumption that the initial matrix contains at least one observation per row and one observation per column. Therefore, to provide more relevant solution. The process adopts an assignment function. This function has as an objective to find the users class that are interested by the selected preferences. It exploits the characteristics of the preferences to enrich the data matrix. It provides as a result the matrix of preferences that are weighted according to users' classes. Then for a certain preference, it is possible to know the class of users that will be the most interested by it. The exact rate will be then equal to the aggregation of rates provided by the users of this cluster.

Algorithm 2: Assignment function to predict the rate of unrated preferences

---

Require : R, C, T,L, Pu, Pi, n, m, u, p,c
- R users-rating matrix
- C preferences-characteristics matrix
- T preferences-users'classes matrix
- L unrated preferences- characteristics matrix
- Pu users partitions
- Pi items partitions
- u number of users

- p number of preferences
- n the number of users clusters
- m number of characteristics
- c number of unrated preferences

Ensure: X

M(u,p);
M=R*C;

for i from 1 to n
do

Cu ← {Cu ϵ Pu : u ϵ Cu} ;
m=size(Cu) ;
for j from 1 to m do
T(i,:)= (T(i,:)+M(i,:))/m;
end for
end for

X(c,p);

X= T*L';

4. Numerical results

The evaluation of the approach is done using the MovieLens dataset. It consists of:

1- 100 000 ratings from 943 users on 1682 films from 1 to 5.
2- Each user has rated at least 20 movies.
3- The data sets are 80% 20% splits into training and test data.

The experiments were conducted via a modest machine with the following characteristics: i7-4510U CPU 2.00GH

We performed the first step of their approach to detect 10 clusters, 5 for users and another 5 for films according to the rating scale. This step has as complexity of O(nkt) where n refers to the number of data objects while t is the number of iteration, k of course is the number of classes generated.

The second step that corresponds to the predictive method allowed them to recover the initial matrix R as the matrix $\tilde{R}$ which dimension is 943x1682 from only 100 000 known data that corresponds to almost 6.5% of global data.

In the objective to demonstrate the efficiency of the combination between the aggregation method and the SVT algorithm per blocks, they applied several methods of Low-Rank Matrix Recovery and Completion over the same experimental data. These methods minimize also the nuclear norm of their users-preferences matrix in the aim to recover the missing data with a precise rank. The researchers cite Augmented Lagrange multiplier method ALM, Accelerated Proximal Gradient method APG[10], Dual Method DM [11] and Fixed-Point Continuation method FPC[12]. Only SVT, FPC and ALM algorithms recovered the matrix with the desired rank 943.

They compared the results obtained according to four metrics: Mean Absolute Error MAE, Root Mean Square Error RMSE, Relative recovery error E_1, Relative recovery in the spectral norm E_2.

- Mean Absolute Error MAE is defined as the average of the absolute error, which is the difference between the predicted rating $\tilde{R}_{ij}$ and the actual rating R_{ij}.

$$MAE = \frac{1}{N}\sum_{k=1}^{N}\left|\tilde{R}_{ij} - R_{ij}\right| \quad (6)$$

- Root Mean square Error RMSE is biased to provide more weights to larger errors.

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(\tilde{R}_{ij} - R_{ij})^2} \quad (7)$$

- Relative recovery error E_1

$$E_1 = \frac{\left\|R - \tilde{R}\right\|_F}{\|R\|_F} \quad (8)$$

- Relative recovery in the spectral norm E2

$$E_2 = \frac{\left\|R - \tilde{R}\right\|}{\|R\|} \quad (9)$$

The dimension reduction of the matrix using the PCA over the filtered data matrix presents better results than the use of the Singular Value Decomposition method SVD.

Table 1: Initial matrix results

		Using the initial matrix r=943			
		MAE	**RMSE**	**E_1**	**E_2**
ACP	**SVT**	1.711969e-01	4.282370e-01	2.683331e-01	5.149617e-01
SVDs		2.269758e-01	6.460150e-01	3.295744e-01	6.323975e-01
SVT		2.348828e-01	6.830165e-01	3.388814e-01	6.429831e-01

FPC	2.470679e-01	7.519525e-01	3.555718e-01	6.671608e-01
ALM	2.500193e-01	8.002567e-01	3.788491e-01	7.34201e-01

Table 2: Filtered matrix results

		Using filtered data r=943			
		MAE	**RMSE**	**E_1**	**E_2**
ACP	**SVT**	1.679444e-01	4.201040e-01	2.683049e-01	5.149614e-01
SVDs		2.229576e-01	6.351916e-01	3.299153e-01	6.354504e-01
SVT		2.343184e-01	6.716209e-01	3.392440e-01	6.438293e-01
FPC		2.447049e -01	7.417087e-01	3.565060e-01	6.685592e-01
ALM		2.488644e-01	7.986859e-01	3.769991e-01	7.329985e-01

Table 3: Execution time

		Execution time in min	
		Using filtered data	**Using initial matrix**
ACP	**SVT**	10.03	16.12
SVDs		8.86	11.14
SVT		20.14	25.06
FPC		38.32	42.18
ALM		45.22	47.96

5. Conclusion

A major limitation of the ontology based information retrieval systems is their inability to deliver pertinent results according to the users preferences. Indeed, they depend on the users' queries, which are insufficient for giving a complete picture about what the users are looking for. In fact, these systems return the same result regardless of who submitted the query. In addition, the same user query is not essentially the same intent.

In this work, we presented a construction profile process that is considered to enrich the user query expressed in SPARQL. It is based on three main steps wish are:

- A learning process to identify users and preferences clusters.
- A predictive method using clusters found in step 1 to predict the unknown data.
- An assignment function to find the ratings of unrated preferences.

The proposed prediction process runs on the assumption that the initial matrix contains at least one observation per row and one observation per column. Therefore, to provide more relevant solution. The process adopts an assignment function. This function has as an objective to find the users class that is interested by the selected preferences. It exploits the characteristics of the preferences to enrich the data matrix. It provides as a result the matrix of preferences that are weighted according to users' classes. Then for a certain preference, it is possible to know the class of users that will be the most interested by it. The exact rate will be then equal to the aggregation of rates provided by the users of this cluster.

Acknowledgment

I acknowledge the support provided by my supervisor Pr. Said RAGHAY and the members of the Laboratory of Applied Mathematics and Computer Science, Faculty of Science and Technology, Cadi Ayyad University, Marrakesh, Morocco.

References

[1] Banouar and S. Raghay, "User profile construction for personalized access to multiple data sources through matrix completion method",IJCSNS, vol. 16, no. 10, pp. 51–57, 2016.
[2] G. Koutrika and Y. Ioannidis, "Personalizing queries based on networks of composite preferences", ACM Transactions on Database Systems, vol 35, pp. 1-50, 2010.
[3] J. Cai , J.E. Candès, and C. Zuowei, "A singular value thresholding algorithm for matrix completion", SIAM Journal on Optimization,vol. 20, no. 4, pp. 1956–1982, 2010.
[4] K. Toh, M. Todd, and R. Tutuncu. SDPT3, "a Matlab software package for semidefinite programming. Optimization Methods and Software", vol 11, no 12, pp. 545–581, 1999.
[5] J. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optimization Methods and Software", vol 11, no 1, pp. 625–653, 1999.
[6] S. Ma, D. Goldfarb, and L. Chen. "Fixed point and Bregman iterative methods for matrix rank minimization", vol. 128, no 1, p. 321-353,2011.

[7] P. L. Combettes and V. R. Wajs., "Signal recovery by proximal forward-backward splitting", Multiscale Model. Simul., vol 4, no 4, pp. 1168–1200 (electronic), 2005.
[8] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration". Multiscale Model. Simul., vol 4, no 2, pp. 460–489 (electronic), 2005
[9] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices," UIUC Technical Report UILU-ENG-09-2215, November 2009.
[10] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast Convex Optimization Algorithms for Exact Recovery of a Corrupted Low-Rank Matrix", UIUC Technical Report UILU-ENG-09-2214, August 2009.
[11] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast Convex Optimization Algorithms for Exact Recovery of a Corrupted Low-Rank Matrix", UIUC Technical Report UILU-ENG-09-2214, August 2009.
[12] E. T. HALE, W. YIN and Y. ZHANG, "Fixed-point continuation for l1-minimization: Methodology and convergence", SIAM Journal on Optimization, vol. 19, no 3, pp. 1107-1130, 2008.

Oumayma BANOUAR: PhD student in the Laboratory of Applied Mathematics and Computer Science, Faculty of Science and Techniques, Cadi Ayyad University, Marrakesh, Morocco.

Said RAGHAY: Professor and researcher in the the Laboratory of Applied Mathematics and Computer Science, Faculty of Science and Techniques, Cadi Ayyad University, Marrakesh, Morocco.