User profile construction for personalized access to multiple data sources through matrix completion method

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 based on a mediation architecture. 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 based on data sources available at the time of interrogation. The purpose of the personalization is to facilitate the expression of user needs. It allows him to obtain relevant information by maximizing the exploitation of his preferences grouped in his profile.
In a mediation architecture context, based on the couple mediator-adapter, the personalization process must consider not only the users' profiles but also the semantic description of data sources defined by mediation requests. The mediator solves the problems associated with heterogeneity while adapters describe the available data sources at the time of interrogation. The key for a successful personalized access to multiple data sources lies in a good construction of the user profile. In our work, we propose a virtual integration system founded on ontologies. These ontologies provide a consensual terminology between the multiple data sources integrated. In addition, we adopt a matrix completion method to ensure an automatic and efficient process for the construction of users' profiles.
Key words:
Personalization, User profile, Matrix completion, Mediation architecture, Interoperability.

1. Introduction

Based on the concept of interoperability, systems of data integration aim to implement a collaborative environment between sources for sharing data and services to respond a user request for information. These systems provide a uniform interface to a multiple, autonomous and heterogeneous data sources. Developers design them for use in areas of closed applications. The users of these applications have to not only know the schemas' sources, but also have to assume that all objects of their universe exist in the sources at the time of use. Therefore, their requests are an exact expression of their need.
However, 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. Consequently, the execution of the same request expressed by different users over a mediation system will necessarily not provide the same results. We will talk here about a personalized access to multiple data sources.
The research work here belongs to the field of content personalization over data sources mediation systems. It proposes a solution based on two major process namely: the rewriting (mediation) process and personalization (enrichment) process. The objective is to interpret the users' intentions expressed in their queries on ones that are more complete. These queries have to take into account their preferences and the data sources description.
The first one is a process that identifies the contributing sources in the execution of the user query. It uses their definitions to reformulate it. The user expresses his query according to the terms of a global schema that procure a transparent access to multiple data sources. The rewriting process transforms it in order to evaluate it on the different data sources schemas. In our work, we used the Local-As-View mediation approach that facilitates the modification over schemas' sources on the contrary of the Global-As-View approach. The use of an ontology-based mediator allow us to purvey information and provides mutual access to heterogeneous knowledge. [1]
The second one is the personalization process. It integrates elements of centre of interests or preferences of the user in his query. Based on [2], the user profile is composed of a set of weighted predicates. The weight of a predicate expresses its relative interest to the user. It is specified by a real number between 0 and 1.
The core of the enrichment process is the phase of profile construction. This step should be the most automatic as possible in the objective to minimize the user interaction with the system.
Our idea here is to apply methods of matrix completion so we can construct the matrix of the user preferences from a sample of data. The objective of this article is to present it. Researchers in mathematics developed these methods, known also as compressive sensing techniques, to recover data from sparse and low-ranked matrices via convex programming techniques.

Manuscript received October 5, 2016
Manuscript revised October 20, 2016

We adapted the Singular Value Thresholding algorithm[3] in the aim to construct the users profiles of our integration system. It minimizes the nuclear norm of a matrix subject to certain types of constraints.
Our work adopted also the Model-Driven Architecture as the realization process. MDA has as its foundation on three complementary ideas: direct representation, automation and open standards. MDA believes that the analysis and design models must be independent from any implementation platform. The MDA approach recommends Unified Model Language UML as the language to use to achieve independent analysis models and design from implementation platforms, which was our choice for modelling.
The remaining of this paper is organised as follows. Section 2 overviews the related works to the field of personalization of user queries. Section 3 dresses the personalised mediation architecture adopted when section 4 and 5 introduces the enrichment and the mediation process. We dedicate a hall section to highlight the process of construction of the users' profiles. Finally, we discuss in the section 7 some of our work results and we conclude by our perspectives for the upcoming work.

2. Related work

The demand for personalized approaches for information access increases. Personalized systems address the overload problem by building, managing and representing information customized for individual users. Research into personalization is ongoing in the fields of information retrieval, artificial intelligence, and data mining. [4] surveys some of the most popular techniques for collecting information about users, representing and building user profile. It concludes that user profile must be dynamic as much as possible and reflect long and short-term preferences.
There are three main construction methods: techniques of keyword profile construction, building semantic network profiles and building concept profiles. No matter which construction method is chosen, it is advised to use the user feedback that requires minimal effort from the system users.
[5] classified the existing information retrieval systems IRS into two main domains.
The first one regroups the classic IRS. They considers that the user request is the sole source of knowledge about the user's information need. However, this resource is often insufficient for describing the user's preferences. It consist usually on a small query that is insufficient for giving a complete accurate picture about the user intent and what he is looking for.
In the other hand, [5] presents the second domain, which includes the recent systems. The researchers here aimed to make the process of information retrieval an easy and clear task. For that, they explored three important concepts: User requirement, Queries and System.
Users tend to ask short queries even when the information needed is complex. The user query reformulation is necessary to satisfy their needs. IRS can achieve this satisfaction only when they have a good modelling of the user profile.
Literature offers many methods and algorithms to construct the user profile. These methods differ from each other according to the specific gaol of the personalization system.
Weighted terms is an important approach used in order to build the user profile. It encloses different techniques such as Boolean weighting or Frequency weighting. The central idea governing all these weighting techniques is that the occurrence of terms within documents , for example, characterises the user profiling to which the document belongs. Hence, the more frequently a term occurs in a class, the more indicative it is of that class. The more often a certain tag is used, the higher the interest of the user in the corresponding topic.
Therefore, the simplest method for creating aggregated data for a user's bookmark collection is to count the occurrence of tags. The result of this computation is a list of tags ranked according to tag popularity.
Other research works adopted the graph theory to construct their users' profiles. The main approach here is to extract the user profile from the profile graph. It is a graph with labelled nodes and undirected weighted edges. Nodes correspond to tags used by users for annotating certain bookmark as an example. Edges represent the semantic relationships between tags. Each time a new tag is used, the system adds a new node to the graph. In addition, each time a new combination of tags is used, the system creates a new edge with a weight of one between the corresponding nodes in the graph.
Right now, it is the construction methods depending on machine learning that are still very promising. These methods aim to construct the users' preferences by predicting them from a small set of observations.
There are methods that use the learning techniques to classify the new profile elements to be added such as [6], [7] and [8]. Moreover, most of these learning techniques to classify the already existing elements in order to remove the ones that have become irrelevant.
The proposed method in [6] allows to classify the profile elements by using association rules and Bayesian networks. The system maintains the relevant concepts. The work in [9] adopted the supervised learning technique by using the K-NN algorithm. The classifier uses labelled users' preferences pool to classify the preferences of each user. In [10], the profile elements are represented as a category hierarchy. Each category represents the knowledge about a domain of user interest and has an energy value. This

increases latter when the user shows interest in the category and decreases by a constant value for each period. Based on the energy value, the system classifies the categories: the system removes categories that have low-energy and categories that have high-energy will persist.[11] proposed a method that aims to detect irrelevant elements from an overloaded user profile at first and secondly, to remove the irrelevant ones in order to obtain a pertinent profile. Thus, the input of this method is an overloaded user profile, which consists of the user's navigation history of one user.
Most works here adopted a supervised or semi-supervised learning. In this article, we adopted an unsupervised method for users' profiles construction. Preferences stocked in these profiles allowed us to enrich the user query before or after its reformulation to be executed over multiple and heterogeneous data sources.

3. The personalized access to multiple data sources process

In our work, the enrichment and the rewriting process are dependent. These two algorithms add predicates to the user query. Profile predicates for enrichment and semantic links for rewriting. As their behaviour depends on the predicates of the query to reformulate, the result will depend on the two algorithms application order.

- Rewriting of the user query enriched process

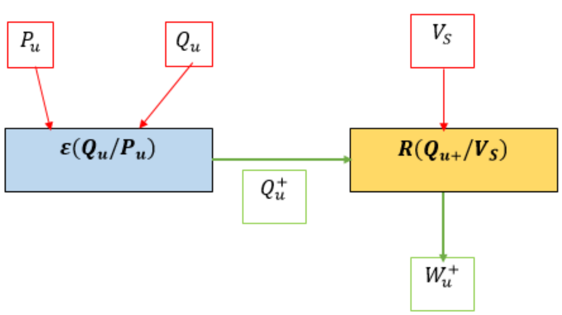

Figure I: Rewriting of the user query enriched process

- Enrichment of the rewritten user query process

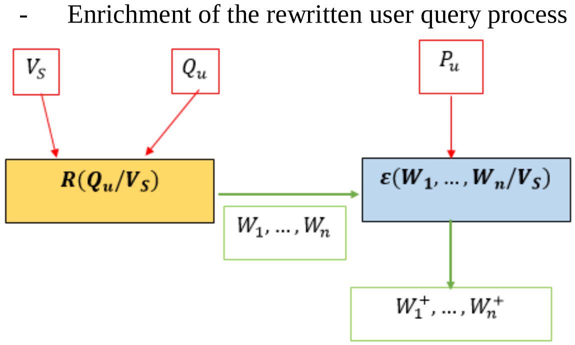

Figure II: Enrichment of the rewritten queries process

V_S: Sources
Q_u: User query
P_u: User profile
W_u: rewriten user query
Q_u^+: enriched user query
W_i^+: enriched and rewritten queries of user query

With:

4. Rewriting process

The query rewriting process depends on the way the system defines mappings. Our system adopts the Local-As-View approach. It defines each sources relationships by a query in terms of the virtual or global schema. This mediation approach facilitates the incorporation of the dynamicity sources. Indeed, changing a source means changing a single query.

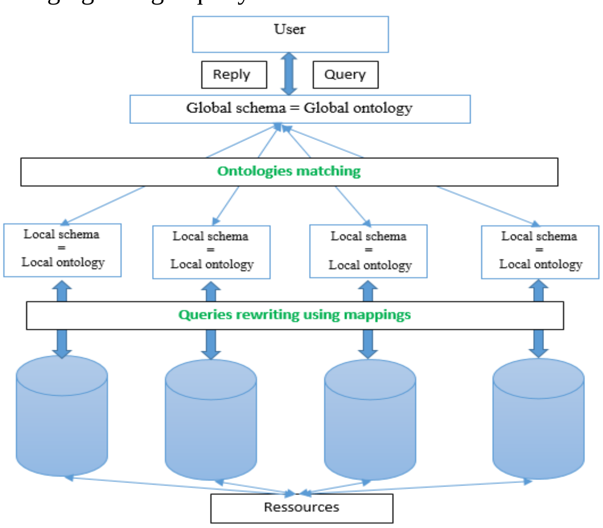

Figure III: Ontology-based mediation process

Each data sources integrated in the system disposes of a local schema describing its structure and content. This schema is obtained through a local ontology. The adoption of a Local-As-View approach for mediation assumes that every data source is defined according to the terms of the global schema procured by a global ontology.
This system free the user from having to locate sources that are relevant to a query, interact with each one in isolation, and combine data from multiple sources. The users do not ask queries in terms of the schemas in which the data is stored, but in terms of a mediated schema. The mediated schema is a set of relations designed for a specific data integration application, and contains the salient aspects of the domain under consideration. The tuples of its relations are not actually stored in the data integration system. Instead, it includes a set of sources

descriptions that provide semantic mappings between the relations in the sources' schemas and the relations in the mediated schema. An ontology-based mediator purveys information and provides mutual access to knowledge.

5. Enrichment process

The approach to personalization consists on maintaining, for every user, a user profile whose structure is related to the features of the data and query models. The personalization system stores preferences at the level of atomic query elements, called therefore atomic preferences. Preferences for values of attributes are stored as atomic selection preferences, and preferences for relationships are stored as atomic join preferences.
It distinguishes between two relevant dimensions of atomic preferences: valence and concern as presented by the following table:

Table 1: Preferences types for enrichment process

Preference type	Principe
Valence	Preferences may be positive: expressing liking, negative: expressing dislike or indifferent: expressing do not care.
Concern	Preferences may be in presence: concerning the presence of values, or absence: concerning the absence of values.

According to [2], for an atomic selection condition q on an attribute A, a user preference for values satisfying (or not) q is expressed by the degree of interest in q, denoted by doi(q), which is defined as follows:

$$doi(q) = < d_T(u), d_F(u) > \qquad (1)$$

where $\forall u \in DA$ satisfying q, $d_T(u), d_F(u) \in [-1, 1]$ and $d_T(u) \times d_F(u) \leq 0$.

The join preferences are simpler as they do not lend themselves to any of the variations mentioned above. A user's preference for a join condition q is expressed by the degree of interest in q, doi(q), defined as follows:

$$doi(q) = < d > \qquad where \qquad d \in [0, 1] \qquad (2)$$

The system builds the implicit preferences that starts from a join preference to a value preference. An implicit query element is the conjunction of the constituent atomic ones. The degree of interest in an implicit preference is the multiplication function of the degrees of interest in the participating atomic preferences.
The atomic and implicit preferences with their degrees compose the preferences predicates. The system orders them and starts the predicates selection process to add to the user query to ensure the enrichment.

This process uses three main parameters:
- K the number of profile predicates that the process should take into account according to their weight.
- M the number of predicates among the K, which must be satisfied: that corresponds to the M predicates largest weight among the top K.
- L the minimum number of predicates from the remaining M - K where each tuple of the result must satisfy.
The enrichment process begins with the extraction of preferences followed by the predicates selection process and their inclusion in the request of the user.
It is clear here, the importance to have a performing process for users profile construction from which the personalization process will extract the preferences and their degrees of interest.

6. User profile construction

In the area of recommender systems, users submit ratings on a subset of entries in a database, and the vendor provides recommendations based on the user's preferences. Because users only rate a few items, one would like to infer their preference for unrated items. Rows of the data matrix represent the system users, when its columns simulate their preferences.
Users are given the opportunity to rate preferences. For sure, they typically rate only very few ones so that there are very few scattered observed entries of this data matrix. Yet one would like to complete this matrix so that the system might recommend information that any particular user is likely will find relevant. In this case, the data matrix of all user-ratings may be approximately low rank because we commonly believe that only a few factors contribute to an individual's tastes or preferences.
[3] considers a problem of considerable practical interest, the recovery of a data matrix from a sampling of its entries. It supposes that the system observes m entries selected from a matrix M. It demonstrates also that one can perfectly recover most low-rank matrices from what appears to be an incomplete set of entries. It proves that if the number m of sampled entries obeys:

$$m \geq Cn^{1.2} \log n \qquad where \qquad n = \max(n_1, n_2) \qquad (3)$$

for some positive numerical constant C, then with very high probability, most n1 x n2 matrices of rank r can be perfectly recovered by solving a simple convex optimization program. This program finds the matrix with minimum nuclear norm that fits the data.
If the number of measurements is sufficiently large, and if the entries are sufficiently uniformly distributed, one might hope that there is only one low-rank matrix with these entries. If this were true, one would want to recover the data matrix by solving the optimization problem:

$$\text{minimize} \quad rank(X) \qquad (4)$$
$$\text{subject to } X_{i,j} = M_{i,j}(i,j) \in \Omega$$

where X is the decision variable, rank(X) is equal to the rank of the matrix X and Ω is the set of locations corresponding to the observed entries: (i,j) Є Ω if Mij is observed.

The program (4) is a common sense approach, which simply seeks the simplest explanation fitting the observed data. If there were only one low-rank object fitting the data, this would recover M. According to [3] this is unfortunately not only a NP-hard problem, but all known algorithms which provide exact solutions require time doubly exponential in the dimension n of the matrix in both theory and practice.

It considers an alternative that minimizes the sum of the singular values over the constraint set. If a matrix has a rank r, then it has exactly r nonzero singular values that the rank function in (4) is simply the number of nonvanishing singular values. The optimization then is given by:

$$\text{minimize} \quad ||X||_* \qquad (5)$$
$$\text{subject to } X_{i,j} = M_{ij}(i,j) \in \Omega$$

where the rank function counts the number of nonvanishing singular values and ||X||* is the nuclear norm that sums their amplitude.

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) Є N.
- b the linear vector which contains the observed elements.
- m_u the smoothing degree.

It resolves the minimization problem of the nuclear norm of X in the objective to complete it. This problem is defined as follow:

$$\text{minimize} \quad ||X||_* + \frac{1}{2}m_u||X||_F^2 \qquad (6)$$
$$\text{subject to } \Omega \times X = b$$

where $||X||_F$ is the Frobenius norm of the matrix X.

7. Results and discussion

The combination of rewriting process with enrichment method for the user query drives us to two scenarios. The first one - rewriting the user query then to enrich it with the user profile predicates E(R) - is more adequate when the objective is to have an efficient reformulation process limiting the number of user profile predicates included to the user query. If instead, the purpose is to satisfy as many preferences as possible, then the second order is the one to adopt, which is the enrichment of the user query then its accommodation according to the sources definitions R(E). The user request is a SPARQL query in terms of the global schema defined by a global ontology.

Table 2: Comparison between the two-application orders of the algorithms

	R(E)	E(R)
contains predicates disjunction	yes	no
decomposes the enriched query	yes	no
integrates all predicates not conflicting with user profile predicates	yes	no
eliminates some candidates rewrites	yes	no
avoids predicates of user profile that already satisfied by the sources definition	no	yes
fixes the relationship schema	no	yes

The cost of enrichment depends on the approach used. In E(R), it is equal to the time required to add predicates of the user profile to the rewriting candidates. Indeed, this approach does not support the expansion of requests (Rewriting candidates) which simplifies enrichment. In R(E), enrichment amounts to: (i) browse the user profile of the graph to find the predicates related to the application and not conflicting with it, (ii) select the Top K profile predicates and (iii) integrating them to the query. The most expensive stage of this process is the stage (i), but it can be done in polynomial time.

The use of the algorithm that take the form of a norm-minimization process allowed us to recover all the information about the users' preferences of our system from some known preferences. The optimisation of the nuclear norm of an n1 x n2 matrix of rank r is based on several terms:

- The number of degrees of freedom noted d_f where $d_f = r(n_1 + n_2 - r)$
- The oversampling ratio k: the ratio between the number of sampled entries m and the number of degrees of freedom d_f defined as: k= m/d_f. The k values are determined by experimentation and should vary as following:

$$C \times n^{1.2} \times \log n \times \frac{1}{d_f} \leq k \leq (n_1 \times n_2 - 1) \times \frac{1}{d_f}$$

where $n = max(n_1, n_2)$ and $C \in R^+$

- The percentage p between the observed entries and the matrix X size where

$$p = \frac{m}{n_1 \times n_2} \times 100$$

The relative recovery error uses the Frobenius norm to give an indication of how good the method adopted recovers and estimates the users' preferences matrix. It is defined as:

Table 3: Matrix completion results

N=size(n_1xn_2)	r_m	d_f	k	r_X	M	P(%)	Time(s)	iterations	E
1000x1000	10	19900	4	16	79600	8	57	200	1.77E-04
1000x1000	10	19900	5	10	99500	10	15	122	1.76E-04
1000x1000	10	19900	6	10	119400	12	13	104	1.50E-04
1000x1000	10	19900	7	10	139300	14	11	92	1.43E-04
1000x1000	50	97500	4	50	390000	39	78	109	1.47E-04
1000x1000	50	97500	2	50	487500	49	71	85	1.41E-04
1000x1000	50	97500	6	50	585000	59	54	71	1.37E-04
1000x1000	50	97500	7	50	682500	69	42	61	1.38E-04
1000x1000	100	190000	4	100	760000	76	96	81	1.49E-04
1000x1000	100	190000	5	100	950000	95	86	52	1.60E-04
5000x5000	10	99900	2	10	599400	3	107	108	1.65E-04
5000x5000	10	99900	7	10	699300	3	92	97	1.52E-04
5000x5000	10	99900	8	10	799200	4	94	88	1.50E-04
5000x5000	10	99900	20	10	1998000	8	96	77	1.38E-04
5000x5000	10	99900	30	10	2997000	12	118	47	1.12E-04
5000x5000	10	99900	40	10	3996000	16	140	43	1.08E-04
5000x5000	50	497500	6	50	2985000	12	611	90	1.48E-04
5000x5000	50	497500	7	50	3482500	14	640	81	1.37E-04
5000x5000	50	497500	8	50	3980000	16	640	74	1.36E-04
5000x5000	50	497500	20	50	9950000	40	638	47	1.12E-04
5000x5000	50	497500	30	50	14925000	60	635	40	1.00E-04
5000x5000	50	497500	40	50	19900000	80	745	35	9.47E-05
5000x5000	100	990000	6	100	5440000	24	1002	84	1.45E-04
5000x5000	100	990000	7	100	6930000	28	1042	75	1.38E-04
5000x5000	100	990000	8	100	7920000	32	1050	66	1.34E-04
5000x5000	100	990000	20	100	19800000	80	1046	64	1.33E-04
5000x5000	100	990000	25	100	24750000	99	1030	60	1.33E-04

$$\varepsilon = \frac{||M - X||_F}{||M||_F} \quad (7)$$

where M is the matrix that we want to recover using the SVT algorithm.

The application of the SVT algorithm for the completion of our matrix that contains the users' evaluation of their preferences presents the following numerical results.

The objective of the SVT algorithm is to find that matrix with the true rank r and which minimize the objective function.

The application of the algorithm over a 1000x1000 matrix with a rank r=10 allows us to find the adequate k (k=5) from which SVT returns the exact rank of the unknown matrix X. The several numerical simulation over 1000x1000 and 5000x5000 matrices demonstrate the performance and the effectiveness of the SVT algorithm for law-rank matrices.

Table 4: Completion matrix results for N=10000x10000

N=size(n_1xn_2)	r_m	d_f	k	m	P(%)	Time(s)	iterations	E
10000x10000	10	199900	6	1199400	1.2	281	123	1.73E-04
10000x10000	10	199900	20	3998000	4	270	110	1.73E-04
10000x10000	10	199900	40	7996000	8	260	44	1.72E-04
10000x10000	10	199900	80	15992000	16	215	37	1.70E-04
10000x10000	10	199900	160	31984000	32	210	35	1.70E-04
10000x10000	10	199900	240	47976000	48	200	30	1.66E-04
10000x10000	10	199900	320	63968000	64	199	26	1.65E-04
10000x10000	10	199900	400	79960000	80	199	22	1.64E-04
10000x10000	10	199900	480	95952000	94	198	20	1.64E-04

Indeed, the algorithm recovers the unknown matrix X with the minimum of time execution and relative error when r=10 compared to r=50 or r=100. The SVT algorithm can recover a matrix representing the rating of 10 000 users for 10 000 preferences with a very law-rank r=10 in few minutes from 1.2% of the matrix X as the number of sampling entries and running in a modest machine. This result proves the performance and the effectiveness of the algorithm when we will use more performing machine and augment the number of sampling entries to an average rate. This algorithm is easy to implement and surprisingly effective both in terms of computational cost and storage requirement when the minimum nuclear-norm solution is also the lowest-rank solution.

8. Conclusion

In order to enable the sharing of resources, ontology-based mediation systems cope with information heterogeneity problems namely semantic and structural conflicts. Ontologies mediates incomprehensible information between several data sources.

A major limitation of this kind of integration 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.

We proposed in this article a solution that combines between two major processes: Enrichment process and Rewriting process. The system reformulates the users' queries by enriching them by users' preferences and it accommodates them to be executed on several data sources.

We presented an automatic process for users' profiles construction. Then we integrated it in a system that provides personalized access to multiple data sources guided by an MDA approach in a LAV mediation architecture context.

The construction process based on a performing matrix completion algorithm, models users as the lines of a matrix X and preferences as its columns. Its execution on a modest machine provides us the possibility to recover the degrees of interest of 10 000 users over 10 000 preferences from only 1.2% of data observed in just very few minutes.

The objective now is to evaluate the performance of this algorithm by running it with more performing machines and to proceed into a classification process before using it.

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 Techniques, Cadi Ayyad University, Marrakesh, Morocco.

References

[1] M. Kupčík, Š. Michal, and B. Zdeněk. "Interoperability through ontologies.", IFAC Proceedings, Vol. 45, no. 7, pp. 196-200, 2012.

[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, C. Zuowei, "A singular value thresholding algorithm for matrix completion.", SIAM Journal on Optimization,Vol. 20, no. 4, pp. 1956–1982, 2010.

[4] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli, "User Profiles for Personalized Information Access.", the adaptive web Springer Berlin Heidelberg , pp. 54–89, 2007.

[5] S. Alaoui, Y. El, B. El Bouzekri El Idrissi, and R. Ajhoun, "Building rich user profile based on intentional perspective.", Procedia - Procedia Comput. Sci., Vol. 73, no. Awict, pp. 342–349, 2015.

[6] L. Chunyan: "User profile for personalized web search.", Fuzzy Systems and Knowledge Discovery FSKD, pp. 1847-1850, 2011.

[7] M. Pazzani,, D. Billsus: "Content-Based Recommendation Systems.", The Adaptive Web, pp. 325-341, 2007.

[8] A. Sieg, B. Mobasher, R. Burke: "Web search personalization with ontological user profiles." ACM conference on Conference on information and knowledge management, pp 525–534 2007.

[9] M. Montaner, B. Lopez, J. De La Rosa: "A Taxonomy of Recommender Agents on the Internet.", Artificial Intelligence Review, Vol. 19, no. 4, pp. 285-330, 2003.

[10] C. Chen, M. Chen, Y. Pva Sun : "A self-adaptive personal view agent." Journal of Intelligent Information Systems, Vol. 18, no. 2-3, pp. 173- 194 2002.

[11] R. Zghal, L. Ghorbel, C. Amel, and I. Amous, "Pertinent user profile based on adaptive semi-supervised learning.",Procedia - Procedia Comput. Sci., vol. 22, no. 1, pp. 313–320, 2013.

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.