

Recommendation Strategy based On Relation Rule Mining

Mehdi Adda¹, Rokia Missaoui², Petko Valtchev¹, Chabane Djeraba³

¹DIRO Université de Montréal, Canada

²Département d'informatique et d'ingénierie, UQO, Canada

³LIFL Université de Lille, France

{addamehd, valtchev}@iro.umontreal.ca, rokia.missaoui@uqo.ca, djeraba@lifl.fr

Abstract

Web users are nowadays confronted with the huge variety of available information sources whose content is not targeted at any specific group or layer. Recommendation systems aim at adapting this content to (their guesses about) the needs of a particular user and hence usually compute some sort of relevance score of the manipulated content objects. As direct information about user needs is scarce, content objects are assessed not directly with respect to those needs but rather in relative manner, i.e., as compared to other objects whose relevance is known. The likeness indices for objects vary from association degrees computed from user logs to inter-object similarities to aggregations of direct user votes on object relevance. We claim that as structured content descriptions, i.e., by means of an ontology, get ever more popular among information providers on the Web, the underlying domain knowledge may successfully be exploited in comparing objects for recommendation purposes. In this paper, we introduce a recommendation approach that explores a specific sort of domain knowledge, the inter-object relational links (e.g., *part-of*, *powered-by*, *same-author-as*, etc.), that are typically expressed at the ontological level by means of specialized languages like OWL. These links form the backbone of a new sort of behavioral patterns, called *relation rules*, that are extracted from user logs. The basic notions, definitions and mining algorithm for relation rules are provided and illustrated by means of sample ontology and content object set of e-commerce flavor.

Keywords

Recommendation systems, personalization, ontology, data mining.

1 Introduction

The Internet has taken a fast growth over the past years and is now playing a central role in information exchanges. The progress allows providers and public administration to offer their products and services directly to a huge population of Web users. However, while the offer is usually rich, only a limited subset of the information items, or content objects, available on a Web site are relevant to the needs/preferences/tastes of a particular user. Recommendation systems are developed with a mission to help users find

items of interest among those served by a Web application. As direct information about user needs is scarce, essentially for confidentiality reasons, relevance is rarely assessed by directly confronting content objects to (a structured representation of) those needs. Instead, items are compared among them with the heuristic guess that likeness will mean potential relevance. Here likeness scores can be obtained by processes as diverging as similarity computation from structured descriptions of items, association extraction from co-occurrences of items in user logs or aggregation of direct user votes on the relevance of particular items.

We claim that regardless of the actual assessment approach for item likeness, higher degrees correlate with the existence of semantic links among item pairs. Hence, in order to increase effectiveness and efficiency of recommendation systems, we propose to directly incorporate knowledge about the existence of such links in the assessment process. Indeed, working with an explicit representation of such links rather than inferring them *a posteriori* and on an individual basis allows for a better, i.e., more precise and at a higher abstraction level, evaluation of regularities in the interactions between users and items or among items themselves.

Ontologies provide the formal framework for expressing knowledge about a domain, comprising semantic links among domain individuals that are described at the conceptual level as inter-concept relations or roles. Ontologies are seen as interoperability means on the Web and therefore a growing number of information providers on the Web power their applications with an ontology describing the content that is served. Hence, it seems reasonable to make a recommendation system rely on a domain ontology in the assessment of item relevance.

We are investigating a recommendation approach that explores an available ontology as source of two types of knowledge about content objects: membership to generic concepts and existence of inter-object relational links. The approach is memory-based as its key component is an analyzer of user logs that reveals patterns of user-item interaction. The discovered regularities are used to choose optimal "next item" for on-line recommendation to a user. Two different types of patterns are mined, class patterns and relation patterns. While the former kind is closely related to what is known as *generalized patterns* in the data mining field [Srikant and Agrawal, 1995], the latter type represents an original notion that has

not been studied so far, at least to the best of our knowledge. As their respective names reveal, class patterns are made of ontology concept (class) names while relation patterns include relation (role) names. Both represent sequences of ontology elements that, once their interestingness established, are matched against a user session to figure out a subset of items to recommend. The combination of both types of patterns provably increases the precision of the recommendation, especially in case of a large population of content objects. To motivate the relation pattern concept, consider a company Web site featuring digital cameras for sale and assume that the company decides to power the site with a domain ontology and a log-analysis-based recommendation system. Imagine that the site users, once they have made up their minds on a purchase of a camera body, typically look for a compatible zoom. An average mining method for plain pattern would easily extract co-occurrences of cameras and zooms and there may be some concrete patterns that become frequent with time and hence get selected for recommendation. As those remain of very local scope, a more sophisticated approach would be to generalize such patterns at the class level, thus leading to a unique pattern roughly saying "recommend zooms after a camera has been targeted". However, if many cameras and zooms are sold on the site, by following this pattern a user picking a camera would be proposed a large number of zooms whereby only few of them would match the actually selected body. The obvious noise could be easily avoided if the pattern could be spelled as: "recommend compatible zooms" which directly refers to a compatibility relation between cameras and zooms. Our study is about the extraction and application of this kind of patterns.

In this paper we lay the foundations of the approach by providing the definitions of key concepts such as relationset, relation association rule, support and confidence of such rules. Moreover, we formulate straightforward algorithms for the related mining tasks. The entire set of novel constructs are illustrated by means of a simple e-commerce flavored ontology which has been adapted from a well-known ontology¹ and further completed with a set of content objects.

The rest of the paper is organized as follows. Section 2 reviews some definitions related to ontology and description logics. Section 3 introduces new notions that are used in the paper, formally defines the approach that we propose, and describes algorithms that we developed to make recommendations from relation rules. Section 4 provides an illustrative example of our approach. Section 5 gives an overview of some related work on personalization and recommendation. Concluding remarks and discussions on future work are given in Section 6.

2 Ontologies and Description logics

An ontology is a conceptual schema expressed in a suitable language which provides necessary constructors to add semantics to the represented information. Ontologies have recently gained interest with the emergence of the Semantic Web [Berners-Lee *et al.*, 2001], and some related standardization efforts are being conducted at several levels and for sev-

eral purposes such as IST OntoWeb [Horrocks *et al.*, 2005], DAML+OIL [Commette, 2001], SemanticWeb.org [Decker and Sintek, 2005], and OWL [McGuinness and van Harmelen, 2004]. Some of them give a meaning to the text while others go further and help make assertions and infer new knowledge.

2.1 OWL

OWL (*Ontology Web Language*) is a standard for domain knowledge representation which is defined to be used as a general structure in the semantic web [McGuinness and van Harmelen, 2004]. OWL allows semantic web ontology to be expressed using concepts and roles and a specific set of connectors using XML syntax. OWL defines three sub-languages from the less expressive one to the more expressive one : OWL-Lite, OWL-DL and OWL-Full. We are interested in this paper by the use of OWL-DL, which implements all the functionalities allowed by the description logic (DL) formalism and is sufficient enough to our work. The basic principles of DL language are given in the following section.

2.2 Description logics

Description logics formalism for knowledge representation and description is mainly characterized by a set of constructors that allow the definition of complex concepts and roles from atomic ones. An atomic concept is an unary relation that can be considered as a class of a set of objects called individuals, whereas an atomic role corresponds to a binary relation between two concepts. Concepts (respectively roles) are called primitives if they are not defined from other concepts (respectively other roles). Concepts and roles grouped together form the terminology and the vocabulary for the application domain and are labelled by the term *TBox*. The second component of a DL knowledge base is called *ABox* which contains factual assertions about individuals based on the *TBox* concepts and roles. For more details about DL, the reader is referred to Baader *et al.* [Baader *et al.*, 2003].

In the remainder of this article we will use OWL terms like class, object and relation to indicate concept, individual and role respectively.

2.3 Example domain ontology in OWL

As an example of OWL ontology, we consider the domain of electronic commerce for which we have extended an existing ontology about cameras. In order to get a richer representation of the underlying domain, we have added concepts and roles to cover more products and accessories. The resulting ontology can be considered as an ontology for a part of real electronic retail site. Figure 1 is a partial view of this ontology, where just *is-a* relations are shown using the *Protégé* platform².

3 Ontology relation-based recommendation approach

Since the existence of large amounts of data represents a potential wealth of information, we use adequate methods

¹<http://protege.stanford.edu/plugins/owl/owl-library/camera.owl>

²Protégé : <http://protege.stanford.edu>



Figure 1: Partial view e-commerce ontology .

for transforming data into meaningful information and useful knowledge. One class of such data is stored in transaction databases from which all items obtained in a single transaction can be retrieved as a unit. The transactions can then be examined to determine what items typically appear together, e.g., which items customers buy together in a database of supermarket transactions. According to domain ontology, items and associations are represented in a way that most likely reflects the human perception on the domain under consideration. By analogy with item transaction in association rule applications, a *relation transaction* is a transaction of individual relations obtained from an item transaction (e.g, *laptop, battery, suitcase*) by replacing items with their corresponding relations (e.g, *poweredBy, carriedBy*) drawn from a domain ontology. In a similar way, we define a *relationset* as a set of relations occurring together on a *relation transaction*. To avoid ambiguity in the use of the term transaction, we call the classic transaction *item transaction* as opposed to *relation transaction*.

Formal definitions of *relationset* and *relation transaction* are as follows. Let $\mathcal{I}_{\mathcal{R}} = \{r_1, r_2, \dots, r_n\}$ be a set of distinct relations of the ontology $\Omega : \mathcal{I}_{\mathcal{R}} \subset TBox(\Omega)$. Let $\mathcal{D}_{\mathcal{R}}$ be a set of relation transactions, where each transaction $\mathcal{T}_{\mathcal{R}}$ is a subset of $\mathcal{I}_{\mathcal{R}}$. A unique identifier is associated with each relation transaction. $X_{\mathcal{R}}$ is called *relationset* if $X_{\mathcal{R}} \subseteq \mathcal{T}_{\mathcal{R}}$. A *relation association rule* is an implication of the form : $X_{\mathcal{R}} \Rightarrow Y_{\mathcal{R}}$, where relationsets $X_{\mathcal{R}}$ and $Y_{\mathcal{R}}$ verify $X_{\mathcal{R}} \cap Y_{\mathcal{R}} = \emptyset$. A *k-relationset* (respectively *k-itemset*) is a relationset (resp. an itemset) composed of k relations (resp. items). The most common measure associated with an itemset (and a rule) is its support, the percentage of all transactions that contain the itemset [Agrawal *et al.*, 1993]. A relation rule of the form $X_{\mathcal{R}} \Rightarrow Y_{\mathcal{R}}$ is associated with a confidence measure which is a ratio between the support of the relationset

$X_{\mathcal{R}} \cup Y_{\mathcal{R}}$ and the support of the relationset $X_{\mathcal{R}}$. The confidence gives the conditional probability of having $Y_{\mathcal{R}}$ when $X_{\mathcal{R}}$ occurs.

3.1 Problem Statement

The statement of *relation rule mining* problem is quite similar to the classical *association rule mining* problem. As firstly stated by Agrawal *et al* [Agrawal and Srikant, 1994], the problem of mining association rules is as follows : given a set of transactions \mathcal{D} , the problem is to generate all association rules that have support and confidence greater than user-fixed thresholds called respectively *minsup* for minimum support and *minconf* for minimum confidence. This problem, as stated, stays valid in our case. Thus, the general problem is to find $\Gamma_{\mathcal{R}}$ a set of all frequent relationsets and then generate related relation rules which will be used to make recommendations. The problem of relation rule based recommendation can be decomposed into three subproblems :

1. **Data transformation** : item transactions are transformed into relation transactions (see Subsection 3.2 for more details).
2. **Frequent relationset mining and association rule generation** : find all relations that have a transaction support above *minsup* fixed by the user. This corresponds to the proportion of transactions that contain a relation. Then, relation association rules with confidence above user-fixed *minconf* are generated. At this step, relation transactions can be considered as item transactions, and any frequent itemset mining algorithm (e.g., Apriori algorithm [Agrawal and Srikant, 1994]) is suitable to mine frequent relationsets. In subsection 3.3 we present an algorithm for frequent relationset and relation rule mining.
3. **Recommendation** : find most relevant relation rules having an antecedent similar to the relations detected be-

tween the last user visited items, and suggest items involved in the consequence of these relation rules. If the set of relation rules is empty, frequent 1-relationships are used to recommend concepts. An algorithm for item recommendation is presented in Subsection 3.4.

3.2 Data transformation: from item transactions to relation transactions

As we mentioned earlier, transactions on original data are sets of items. Therefore, we need to transform them into relation transactions in order to further generate k-relationships and relation rules. A solution consists to exploit the underlying ontology structure of domain knowledge and construct relation transactions as shown by Algorithm 1, where each item transaction is handled separately by the first loop (line 4). After that, a second loop (line 6) iterates over items and for each one of them, we construct the set of all ontology relations in which that item is involved in. Then, we select only relations where at least one of the remaining items on the current item transaction is involved in (lines 9 to 15).

Algorithm 1 iTransacTrans : transforming item transactions to relation transactions

```

1:  $\mathcal{D}$  : set of all item transactions
2:  $\Omega$  : domain ontology
3:  $\mathcal{D}_{\mathcal{R}} \leftarrow \emptyset$ ; //set of all relation transactions

4: for all  $\mathcal{T} \in \mathcal{D}$  do
5:    $\mathcal{T}_{\mathcal{R}} \leftarrow \emptyset$ ; // empty relation transaction
6:   for all  $i \in \mathcal{T}$  do
7:      $\mathcal{T} \leftarrow \mathcal{T} - \{i\}$ ;
8:      $R \leftarrow \text{relations}(\Omega, \{i\})$ ; //get ontology relations
       //in which  $i$  appears
9:     for all  $i' \in \mathcal{T}$  do
10:       $\mathcal{T}_{\mathcal{R}} \leftarrow \mathcal{T}_{\mathcal{R}} \cup \{r/r \in R \wedge (i, r, i') \in \Omega\}$ ; // r
       //connects  $i$  and  $i'$  in  $\Omega$ 
11:       $R \leftarrow R - \mathcal{T}_{\mathcal{R}}$ ;
12:      if  $R = \emptyset$  then
13:        break;
14:      end if
15:    end for
16:  end for
17:   $\mathcal{D}_{\mathcal{R}} \leftarrow \mathcal{D}_{\mathcal{R}} \cup \{\mathcal{T}_{\mathcal{R}}\}$ ;
18: end for
19: return  $\mathcal{D}_{\mathcal{R}}$ ;

```

3.3 Frequent relationsets and relation rules generation

To mine frequent relationsets within a set of relation transactions, Algorithms 2 and 3 are used, where the first one is used to mine frequent 1-relationships. These algorithms adopt an *Apriori* approach [Aggarwal *et al.*, 1999; Agrawal and Srikant, 1994; 1995; Ayres *et al.*, 2002; Han *et al.*, 2000; Leleu *et al.*, 2003] to mine frequent relationsets by candidate generation. As mentioned earlier, other approaches can be used to mine frequent relationsets. Once all frequent relationsets are generated, the *ruleMiner* procedure is called, which

generates *relation rules* with confidence above the user-fixed *minconf*. The generated rules are then used as input in Algorithm 4 to select item that will be recommended to the current user.

Algorithm 2 Mining Frequent 1-relationships

```

1:  $\mathcal{D}_{\mathcal{R}}$  : set of all relation transactions
2: minsup : minimum support;
3:  $\Gamma_{\mathcal{R}}^1 \leftarrow \emptyset$ ;
4:  $\mathcal{R} = \cup_{\mathcal{T}_{\mathcal{R}} \in \mathcal{D}_{\mathcal{R}}} \mathcal{T}_{\mathcal{R}}$ ; //R set of all  $\mathcal{D}_{\mathcal{R}}$  relations

5: for all  $r \in R$  do
6:    $n \leftarrow 0$ ; // n : number of occurrences
7:   for all  $\mathcal{T}_{\mathcal{R}} \in \mathcal{D}_{\mathcal{R}}$  do
8:     if  $r \in \mathcal{T}_{\mathcal{R}}$  then  $n \leftarrow n + 1$ ;
9:   end if
10:  end for
11:  if  $n \geq \text{minsup} \times |\mathcal{D}_{\mathcal{R}}|$  then
12:     $\Gamma_{\mathcal{R}}^1 \leftarrow \Gamma_{\mathcal{R}}^1 \cup \{r\}$ ;
13:  end if
14: end for
15: return  $\Gamma_{\mathcal{R}}^1$ ;

```

3.4 Recommendation

The recommendation algorithm *RAR* (see Algorithm 4) suggests items according to last visited items and to the set of relation rules generated in the previous step. The steps of *RAR* are as follows : (i) select from the domain ontology all relations that hold for the last visited items, (ii) select rules having those relations as antecedent, (iii) recommend new items which are involved in at least one rule consequence.

4 An illustrative example

In order to show the need for broadening the recommendation process by using domain ontology relations in the personalization process, we introduce an example that illustrates the way relation recommendation strategy can deal with some cases better than what previous recommendation approaches might do. For simplicity reasons, we limit ourselves to recommendations produced from 1-relationships. Due to the lack of web sites where content is described using domain ontology, we produced a synthetic dataset to illustrate our approach. The dataset corresponds to virtual user transactions of an electronic retail site. The site content covers a variety of digital products and accessories, such as digital cameras, batteries, mp3-players, etc. We have also developed an OWL ontology presented in Subsection 2.1 to describe the site content and semantic relations holding among products. The dataset contains eleven transactions.

Let Ω be the ontology of Figure 1 that describes the content of our virtual site. Let L be a set of Ω concepts and D the set of item transactions as indicated in Figure 2. In Figure 3 we replace each item by a unique identifier so that transactions will be handled easily.

We then use Apriori algorithm to generate all frequent itemsets (see Figure 4), where the corresponding support is greater than the *minsup* fixed to 0.1.

Algorithm 3 FrM : Mining Frequent relationsets and relation rules generation

```

1:  $\mathcal{D}_{\mathcal{R}}$  : set of all relation transactions
2:  $minsup$  : minimum support;
3:  $minconf$  : minimum confidence;
4:  $\Gamma_{\mathcal{R}}^1 \leftarrow$  all frequent 1-relationsets;

5: for  $k = 2; \Gamma_{\mathcal{R}}^{k-1} \neq \emptyset; k++$  do
6:    $P^k \leftarrow candidateGen(\Gamma_{\mathcal{R}}^{k-1});$  // k-relationset
   //candidates generation from  $\Gamma_{\mathcal{R}}^{k-1}$ 
7:   for all  $t \in \mathcal{D}_{\mathcal{R}}$  do
   //Add candidates contained in  $t$ 
8:      $P_t^k \leftarrow subset(P^k, t);$ 
9:   end for
10:  for all  $c \in P_t^k$  do
11:     $count(c)++;$ 
12:  end for
13:   $\Gamma_{\mathcal{R}}^k = \{c \in P_t^k / count(c) \geq minsup \times |\mathcal{D}_{\mathcal{R}}|\};$ 
14: end for

15:  $\Gamma_{\mathcal{R}} \leftarrow \cup_k \Gamma_{\mathcal{R}}^k;$ 

16: return (ruleMiner( $\Gamma_{\mathcal{R}}$ ));

17: procedure ruleMiner( $S$  : set of relationsets)
18:  $\mathcal{AS} \leftarrow \emptyset;$  //  $\mathcal{AS}$  : set of relation rules
19: for all  $X \in S$  do
20:   for all  $A$  nonempty subset of  $X$  do
21:      $B = X - A;$ 
22:     if ( $support(AB) / support(A) \geq minconf$ ) then
23:        $\mathcal{AS} \leftarrow \mathcal{AS} \cup \{A \Rightarrow B\};$ 
24:     end if
25:   end for
26: end for
27: return ( $\mathcal{AS}$ );
28: end procedure

```

Item Transaction ID	Transaction Item's label
001	{JumpGear, JumpDrive, Mp3-Case, Ultra-Cord-Earphone}
002	{CyberShot-DSC-P41, MCM-AA-1800mAh-R, 512MO-SecureDigital, Pelican1120, MCM-AA-Charger}
003	{Canon-A95, 128-CompaqFlash, Energizer-AA-2300mAh-R, DC-RC1-Charger}
004	{Sony-Ericson-Cellular, SE-650mAh-BST-30-R, SE-Charger}
005	{eMachine-M34-Laptop, 1GB-3200DDR, Energizer-M35-R}
006	{S-AAA-2300mAh, S-AAA-Charger}
007	{iPod, iPod-Leather-Case}
008	{CyberShot-DSC-P93, MCM-AA-1800mAh-R}
009	{JumpGear, JumpDrive, Mp3-Case}
010	{CD-RW-Pack, Mp3-Case}
011	{Energizer-AA-2300mAh-R, DC-RC1-Charger}

Figure 2: Label of items.

Algorithm 4 RA_r : Recommendation Algorithm

```

1:  $\Omega$  : domain ontology
2:  $\mathcal{C} \leftarrow \emptyset;$  // set of items to recommend
3:  $\mathcal{AS}$  : set of all relation rules generated by FrM;
4:  $I$  : set of last visited items
5:  $\mathcal{SR} \leftarrow \emptyset;$  // set of selected rules
6:  $\mathcal{R} \leftarrow relations(\Omega, I);$  // set of the ontology relations
   involved in  $I$ 

7:  $\Gamma_{\mathcal{R}}^1$  : set of all frequent 1-relationsets;

8: if  $\mathcal{AS} \neq \emptyset$  then
9:   for all  $as \in \mathcal{AS}$  do
10:    if  $antecedent(as) \subseteq \mathcal{R}$  then
11:       $\mathcal{SR} \leftarrow \mathcal{SR} \cup \{as\};$ 
12:    end if
13:   end for

14:  $\mathcal{R} \leftarrow consequences(\mathcal{SR});$  // get consequences of
   selected rules

15: else//use frequent 1-relationsets
16:    $\mathcal{R} \leftarrow \Gamma_{\mathcal{R}}^1;$ 
17: end if

18: for all  $r \in \mathcal{R}$  do
19:    $\mathcal{C} \leftarrow \mathcal{C} \cup items(\Omega, r);$  // get items involved in the
   relation  $r$ 
20: end for
21:  $\mathcal{C} \leftarrow \mathcal{C} - I;$  // keep only not visited items
22: return  $\mathcal{C};$ 

```

Item Transaction ID	Transaction Item's ID
001	{1, 2, 3, 4, 5}
002	{6, 7, 8, 9, 10}
003	{11, 12, 13, 14}
004	{15, 16, 17}
005	{18, 19, 20}
006	{21, 22}
007	{23, 24}
008	{25, 7}
009	{1, 2, 3}
010	{26, 27}
011	{13, 14}

Figure 3: ID of items.

1-Itemset	Support	2-Itemset	Support
{1}	0.18	{1, 2}	0.18
{13}	0.18	{2, 3}	0.18
{7}	0.18	{1, 3}	0.18
{2}	0.18	{13, 14}	0.18
{3}	0.18	3-Itemset	Support
{14}	0.18	{1, 2, 3}	0.18

Figure 4: Frequent itemsets.

To produce recommendations, association rules must be generated. To that end, we need to generate all frequent itemsets. Figure 5 shows association rules produced from itemsets of Figure 4.

2-Itemset	Association Rule	3-Itemset	Association Rule
{1, 2}	1 → 2 2 → 1	{1, 2, 3}	1 → 2,3 2 → 1,3 3 → 1,2
{1, 3}	1 → 3 3 → 1		1,2 → 3 2,3 → 1 1,3 → 2
{2, 3}	2 → 3 3 → 2		
{13, 14}	13 → 14 14 → 13		

Figure 5: Discovered association rules.

To get the set of 1-relationsets and make recommendations, we use algorithms mentioned earlier. First, we generate relation transactions using Algorithm 1. The resulting output is given in Figure 6.

Transaction Item's label	Transaction Relation's label
{JumpGear, JumpDrive, Mp3-Case, Ultra-Cord-Earphone}	{extendedMemory, carry, listenWith}
{CyberShot-DSC-P41, MCM-AA-1800mAh-R, 512MO-SecureDigital, Pelican1120, MCM-AA-Charger}	{poweredBy, extendedMemory, carry, chargedWith}
{Canon-A95, 128-CompaqFlash, Energizer-AA-2300mAh-R, DC-RC1-Charger}	{poweredBy, extendedMemory, carry, chargedWith}
{Sony-Ericson-Cellular, SE-650mAh-BST-30-R, SE-Charger}	{ poweredBy, chargedWith }
{eMachine-M34-Laptop, 1GB-3200DDR, Energizer-M35-R}	{poweredBy, extendedMemory }
{S-AAA-2300mAh, S-AAA-Charger}	{chargedWith}
{iPod, iPod-Leather-Case}	{carry}
{CyberShot-DSC-P93, MCM-AA-1800mAh-R}	{poweredBy}
{JumpGear, JumpDrive, Mp3-Case}	{extendedMemory, carry}
{CD-RW-Pack, Mp3-Case}	{ }
{Energizer-AA-2300mAh-R, DC-RC1-Charger}	{ chargedWith }

Figure 6: From item transactions to relation transactions.

Relation Transaction ID	Transaction Relation's label
100	{ extendedMemory, carry, listenWith }
101	{ poweredBy, extendedMemory, carry, chargedWith }
102	{ poweredBy, extendedMemory, carry, chargedWith }
103	{ poweredBy, chargedWith }
104	{ poweredBy, extendedMemory }
105	{ chargedWith }
106	{ carry }
107	{ poweredBy }
108	{ extendedMemory, carry }
109	{ chargedWith }

Figure 7: Relation transactions with labels.

We note that there is no relation transaction generated from the item transaction with $ID = 010$. This is due to the fact that the corresponding items are not directly related in the considered ontology. Then, we associate a unique ID to each relation transaction as shown in Figure 8. At this stage, Algorithm 2 generates all frequent 1-relationsets where the corresponding support is greater than the user-fixed $minsup$. We keep the same $minsup$ as earlier ($minsup = 0,1$). Once the

generation phase is over, Algorithm 4 produces recommendations according to the items that are freshly visited. Recommendations consist on a set of ontology concepts (items) involved in the selected 1-relationsets. Figure 9 gives some examples of recommended concepts according to the last item visited.

Even with simple relationsets, the example shows that recommendations made from relations are more precise and interesting than those produced using classical recommendation approaches.

Relation Transaction ID	Transaction Relation's ID	1-Relationset	Support
100	{a, b, c}	{a}	0.5
101	{d, a, b, e}	{b}	0.5
102	{d, a, b, e}	{c}	0.1
103	{d, e}	{d}	0.5
104	{d, a}	{e}	0.5
105	{e}		
106	{b}		
107	{d}		
108	{a, b}		

Figure 8: Relation transactions with ID; support of 1-relationsets.

Last item visited	Recommendations		
	Classic approaches	Relation-based approach	
		Item-based	RAR selected relations
Digital Camera Concord Eye-Q 4363Z	-	poweredBy extendedMemory	AA Batteries Secure Digital memories
Logitech Cordless Optical Mouse	-	poweredBy	Camera cases AA Batteries
HP iPAQ hx4700 Pocket PC	-	extendedMemory	Secure Digital, Compact Flash Type II, and Multi Media Card memories

Figure 9: Recommendations.

One of the most known problems in recommendation systems is the new item problem since once new items are added, the system cannot recommend them. This problem is partially addressed with generalized association rules, where new items can be suggested if corresponding concepts are known or at least one of their high-level concepts is known. However, if the concept related to the new item belongs to a different concept hierarchy (taxonomy), it is not possible to produce recommendation with the generalized association rule approach. In such situations, relation rules are able to recommend items even though they are not seen before or do not belong to a given concept hierarchy involved in the rules. New concepts are recommended in relation rules just according to the way they are connected with other known items in the domain ontology.

5 Related work

Most of recent research efforts in personalization and recommendation systems have focused on determining which groups of items, called itemsets, frequently appear together in transactions to make recommendations from usage data

only. From any itemset an association rule may be derived which, given the occurrence of a subset of the items in the itemset, predicts the probability of the occurrence of the remaining items (e.g., [Agrawal *et al.*, 1993] [Srikant *et al.*, 1997], [Graef and Schaefer, 2001], [Lin *et al.*, 2000]). According to the recommendation strategy, two main approaches can be distinguished, (i) Content-based recommendations which are based on item similarities (e.g., [Meteren and Someren, 2000; Yu *et al.*, 2003]), (ii) collaborative filtering where recommendations are produced by computing the similarity between different users' preferences (e.g., [Shardanand and Maes, 1995; Kleinberg and Sandler, 2003; Breese *et al.*, 1998; Sarwar *et al.*, 2001; Yu *et al.*, 2002; 2003]).

However, limited research studies are conducted on developing methods and approaches to integrate knowledge associated with the content in the personalization and recommendation process. Some algorithms have been proposed for finding generalized itemsets from items that are classified by one or more taxonomic hierarchies (e.g., [Han and Fu, 1995], [Srikant and Agrawal, 1995], [Srikant and Agrawal, 1996]). These algorithms are designed to support structured data which are organized into hierarchies.

Dai *et al.* [Dai and Mobasher, 2004] have proposed a general framework for domain knowledge integration at different steps of the personalization process without specifying how this is actually done.

6 Discussion and future work

We proposed and developed a new recommendation approach that extends the classical association rule mining problem from item (concept) to relation by providing the capability to handle ontology domain relationships. We developed algorithms for (i) transforming item transactions to relation transactions, (ii) discovering relation rules from frequent relation-sets, and (iii) producing recommendations.

The approach is able to recommend items even when item transactions are sparse, and is suitable for mining interesting patterns even when item co-occurrences are low. The dataset used in this work is synthetic. Real-life data are currently being collected to validate the proposed approach.

The relationset mining process can be improved by taking into account indirect and implicit item relationships. By doing so, the number of non projected item transactions will decrease. Another extension consists to exploit this approach for site restructuring by making the classes involved in frequent relationsets closer to the user's navigation view. Finally, a deep exploration and development of *relation rule mining* will be helpful not only in personalization applications such e-commerce and ITS, but also in other fields and domains where an ontology is associated to the data.

Acknowledgments

We are grateful to NSERC and Canadian Heritage for their financial support.

References

- [Aggarwal *et al.*, 1999] C. C. Aggarwal, J. L. Wolf, K. Wu, and P. S. Yu. Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. *Fifth ACM SIGKDD international conference on Knowledge discovery and data mining, California, United States*, 1999.
- [Agrawal and Srikant, 1994] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *In Proc. 20th Int. Conference on Very Large Data Bases, VLDB94*, 1994.
- [Agrawal and Srikant, 1995] R. Agrawal and R. Srikant. Mining sequential patterns. *Proceedings of the Eleventh International Conference on Data Engineering, IEEE Computer Society.*, pages 3–14, 1995.
- [Agrawal *et al.*, 1993] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.
- [Ayres *et al.*, 2002] J. Ayres, J. Gehrke, T. Yiu, and J. Flannick. Sequential pattern mining using a bitmap representation. *In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, July 2002*.
- [Baader *et al.*, 2003] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. The description logic handbook : Theory, implementation and applications. *Cambridge University Press*, page 574, 2003.
- [Berners-Lee *et al.*, 2001] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.
- [Breese *et al.*, 1998] J. S. Breese, D. Hackermann, and C. KADIE. analysis of predictive algorithms for collaborative filtering. *Proceeding of the fourteenth conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.
- [Commette, 2001] DAML Joint Commette. Daml+oil language. <http://www.daml.org/2001/03/daml+oil-index.html>, March 2001.
- [Dai and Mobasher, 2004] H. Dai and B. Mobasher. Integrating semantic knowledge with web usage mining for personalization. *Integrating Semantic Knowledge with Web Usage Mining for Personalization, Proceedings of the AAAI 2004 Workshop on Semantic Web Personalization (SWP'04)*, July 2004.
- [Decker and Sintek, 2005] S. Decker and M. Sintek. Semanticweb. <http://www.semanticweb.org>, 2005.
- [Graef and Schaefer, 2001] G. Graef and C. Schaefer. Application of art2 networks and self-organizing maps to collaborative filtering. *7th International Workshop on Open Hypermedia Systems, Austria*, 2001.
- [Han and Fu, 1995] J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. *Proc. of International Conf. on Very Large Data Bases VLDB'95, Zürich, Switzerland*, pages 420–431, 1995.

- [Han *et al.*, 2000] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *ACM SIGMOD Intl. Conference on Management of Data*, 2:1–12, 2000.
- [Horrocks *et al.*, 2005] I. Horrocks, F. van Harmelen, and P. Patel-Schneider. Ontology-based information exchange for knowledge management and electronic commerce. <http://ontoweb.ontoware.org/>, 2005.
- [Kleinberg and Sandler, 2003] J. Kleinberg and M. Sandler. Convergent algorithms for collaborative filtering. *Convergent Algorithms for Collaborative Filtering, 4th ACM conference on electronic commerce.*, 2003.
- [Leleu *et al.*, 2003] M. Leleu, C. Rigotti, J-F. Boulicaut, and G. Euvrard. Go-spade: mining sequential patterns over datasets with consecutive repetitions. *Proceedings of the IAPR Third International Conference on Machine Learning and Data Mining in Pattern Recognition MLDM'03, Leipzig (Germany)*, pages 293–306, July 2003.
- [Lin *et al.*, 2000] W. Lin, S. A. Alvarez, and C. Ruiz. Collaborative recommendation via adaptive association rule mining. *Workshop on Web Mining for E-Commerce- Challenges and Opportunities, United States*, 2000.
- [McGuinness and van Harmelen, 2004] D. L. McGuinness and F. van Harmelen. Owl web ontology language overview. <http://www.w3.org/TR/owl-features/>, 2004.
- [Meteren and Someren, 2000] R.V. Meteren and M.V. Someren. Using content-based filtering for recommendation. *Machine Learning in the New Information Age MLnet / ECML2000 Workshop, Spain*, 2000.
- [Sarwar *et al.*, 2001] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. *Tenth international conference on World Wide Web, Hong Kong*, 2001.
- [Shardanand and Maes, 1995] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating "word of mouth". *Proceedings of the SIGCHI conference on Human factors in computing systems, USA*, 1995.
- [Srikant and Agrawal, 1995] R. Srikant and R. Agrawal. Mining generalized association rules. *In Proceedings of the 21st Int'l Conference on Very Large Databases (VLDB95), Zurich, Switzerland.*, September 1995.
- [Srikant and Agrawal, 1996] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. *Proc. 5th Int. Conf. Extending Database Technology, EDBT, Avignon, France*, 1057:3–17, 1996.
- [Srikant *et al.*, 1997] R. Srikant, Q. Vu, and R. Agrawal. Mining association rules with item constraints. *Proc. of the 3rd International Conference on Knowledge Discovery in Databases and Data Mining, Newport Beach, California*, 1997.
- [Yu *et al.*, 2002] K. Yu, X. Xu¹, J. Tao, M. Ester, and H.-P. Kriegel. Instance selection techniques for memory-based collaborative filtering. *2nd SIAM International Conference on Data Mining, United States*, 2002.
- [Yu *et al.*, 2003] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H.-P. Kriegel. Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering (TKDE), special issue on Mining and Searching the Web*, 2003.