

Semantic Translation for Rule-Based Knowledge in Data Mining

Dejing Dou, Han Qin, and Haishan Liu

Computer and Information Science Department
University of Oregon
Eugene, Oregon 97403, USA
{dou,qinhan,ahoyleo}@cs.uoregon.edu

Abstract. Considering data size and privacy concerns in a distributed setting, it is neither desirable nor feasible to translate data from one resource to another in data mining. Rather, it makes more sense to first mine knowledge from one data resource and then translate the discovered knowledge (models) to another for knowledge reuse. Although there have been successful research efforts in knowledge transfer, the *knowledge translation* problem in the semantically heterogeneous scenario has not been addressed adequately. In this paper, we first propose to use Semantic Web ontologies to represent rule-based knowledge to make the knowledge computer “translatable”. Instead of an inductive learning approach, we treat knowledge translation as a deductive inference. We elaborate a translation method with both the forward and backward chaining to address the *asymmetry of translation*. We show the effectiveness of our knowledge translation method in decision tree rules and association rules mined from sports and gene data respectively. In a more general context, this work illustrates the promise of a novel research which leverages ontologies and Semantic Web techniques to extend the knowledge transfer in data mining to the semantically heterogeneous scenario.

1 Introduction

Information resources distributed across the Internet present structurally and semantically heterogeneous data that are hard to process automatically for knowledge acquisition. These resources include online databases, web services and the Semantic Web [6]. They provide a unique and challenging opportunity for knowledge acquisition in new and meaningful ways. Although standards such as SQL, XML, and OWL [1] reduce the *syntactic* diversity, it is unreasonable to expect schemas or ontologies that describe the *structure* and *semantics* of data to be few in number [7]. A variety of heterogeneity has been observed in different data analysis tasks [17]. It will be extremely helpful for data analysts to have a system that can automatically reuse the knowledge mined from one data resource to another. The traditional data mining solution to the semantic heterogeneity is to first apply data translation or data integration as a pre-processing step to either translate data from one schema to another or integrate data from several

resources into a centralized location (e.g., a data warehouse). However, the data translation or integration approach is not ideal because of following two reasons: (i) communication and storage costs make transferring huge volumes of data infeasible; (ii) individual sites want to maintain privacy (e.g., SSN) and do not want to share this information with other parties.

Knowledge translation is a method to overcome semantic heterogeneity by translating knowledge from one data resource (source) to another (target) to make the knowledge reusable or comparable. It is closely related to *knowledge transfer* or *transfer learning* which focuses on applying knowledge gained in solving one problem to a different but related problem. However, to make existing knowledge transfer algorithms work, it is necessary that the target domain has enough or at least auxiliary training data. It may not be true in the real world problems. For example, a new credit card company without historical data wants to use the classification model generated by its collaborative credit card company to determine whether applicants to this new company are qualified or not. The new company and its collaborative company may use different schemas to store their applicants' data. One way is to translate the applicants' data of the new company to the schema of its collaborator's and use the collaborator's classification model to conduct prediction. However, due to communication costs and privacy issues, it is plausible to translate only knowledge described using its collaborator's schema to the new company's schema without carrying out data translation. It is surprising that little research in knowledge transfer has been done in the semantically heterogeneous scenario where two data resources have different but semantically related representations. For example, the semantic heterogeneities between the attributes describing two credit company databases may include synonyms (e.g., salary vs. pay), subsumption (e.g., graduate_student_status vs. student_status) or functional (e.g., concatenation of customer_firstname and customer_lastname vs. customer_fullname) and so forth. It is hard to say that we can directly apply knowledge transfer algorithms to solve the *knowledge translation* problem in the semantically heterogeneous scenario.

To solve the knowledge translation problem, we first identify two critical theoretical challenges need to be addressed: i) there is no standard formal language to represent semantics of mined knowledge. Previous data mining research in exploring mined knowledge mainly focuses on visualization for aiding human comprehension. Although some XML-based languages, e.g., Predictive Model Markup Language (PMML), have been developed to represent a standard syntax for mined knowledge, few work has been done for helping computers "understand" the semantics of knowledge automatically. We need to represent the knowledge in a way that is computer "translatable" from the source to the target if the source and target use different but semantically related schemas. And ii) what criteria can justify a *correct* translation remains a theoretically hard problem when we try to design algorithms for knowledge translation.

To address these two challenges, in this paper, after introducing more related work (Section 2), we propose to use the standard Semantic Web ontology languages to formally represent the rule-based knowledge, such as decision tree

rules and association rules (Section 3). Ontologies, which have been used for formal specification of conceptualization in traditional knowledge engineering and the emerging Semantic Web, will be used for representing IF-THEN data mining rules. We then give definition of correctness and completeness of knowledge translation based on the soundness of formal inference. We point out the asymmetry of translation based on our definitions. The conditions of rules can be translated from the source to the target by backward chaining with generalized modus ponens. The conclusions of rules can be translated from the source to the target by forward chaining (Section 4). We show the effectiveness of our knowledge translation method through two case studies in real world data (Section 5). We discuss the uncertainty issues in knowledge translation and its extension to distributed data mining as our future work (Section 6). We finally conclude the paper with our contributions in Section 7.

2 Related Work

Knowledge transfer and distributed data mining are major research areas which also deal with heterogeneous data and knowledge resources in data mining.

Knowledge transfer focuses on applying knowledge gained in solving one problem to a different but related problem. It is also treated as one kind of knowledge reuse. Gao *et al.* [15] proposed a locally weighted ensemble framework to combine multiple models. Gupta *et al.* [16] presented an algorithm for leveraging heterogeneous data resources and algorithms using different algorithms for each knowledge resource (e.g., Wikipedia). Eaton [14] proposed a method on learning across multiple resolutions of input and applied this technique to the problem of knowledge transfer in multitask learning. However, most knowledge transfer researches in data mining focus on the source and target data resources with different models, structures or distributions. None of them has discussed the semantic heterogeneities, such as synonyms, subsumption and functions, between the source and target. Also, previous knowledge transfer research does not handle the scenario that there is no data in the target resource.

In distributed data mining (DDM), the heterogeneity that has been studied is mainly focused on the scenario where only incomplete knowledge can be observed at each local site [19]. It is also termed as *vertical fragmentation* in DDM literature [8,20]. Although vertical fragmentation is not the same as the semantically heterogeneous scenario we focus on in this paper, some previous experience is helpful. Of particular interest, Caragea and colleagues [8] used attribute value taxonomy (AVT) to represent the semantics of local data resources. A user ontology and a set of simple interoperation constraints (e.g., equivalence and subsumption) between the data resource ontology and user ontology are first specified manually. Then mappings between local ontologies and the user ontology can be derived and used to answer statistical queries. In the paper, we prefer to use mappings to translate the generated knowledge instead of only translating the statics of queries.

3 Formal Representation of Rule-Based Knowledge

To formally represent mined knowledge, it is important to understand the semantics of the knowledge. We basically borrow the ideas of the Semantic Web [6] which targets at making the web data computer “understandable” and sharable among software agents.

Different data mining tools (e.g., Weka [22], XLMiner [4]) may use different syntax and semantics to represent results (e.g., decision trees). The previous data mining research has focused on the visualization of mined knowledge to help human understanding. However, to enable automatic knowledge translation, we represent the knowledge in a formal way that computers can “understand” and process. To that end, we choose ontologies, which are formal specifications of conceptualization, to formally describe the mined knowledge (e.g., decision tree rules and association rules). Specifically, we leverage the research progress in the Semantic Web on Web Ontology Language (OWL [1]) and one of its related rule languages, SWRL [3], to help formally represent data mining knowledge. Both OWL and SWRL are based on fragments of first order logic (i.e., Description Logics and Horn Logic).

Ontologies can be designed by domain experts manually or be mined from data resources semi-automatically [10]. Once the ontologies are constructed, the classes, properties and axioms of ontologies will be used to describe the knowledge discovered from that data resource. For example, one decision tree rule from a credit card company data can be represented in SWRL. The SWRL uses OWL/RDF syntax which is space consuming. To save the space, we use the general first order logic (FOL) syntax to represent SWRL rules in the paper:

$$\forall x, y \text{ Applicant}(x) \wedge \text{age}(x, y) \wedge (y > 30) \wedge \text{grad_student}(x, \text{“Y”}) \rightarrow \text{credit}(x, \text{“Good”})$$

where “Applicant” is an OWL class (i.e., unary predicate), “age”, “>”, “graduate_student”, and “credit” are OWL properties (i.e., binary predicates). This rule means: “IF an applicant’s age is larger than 30 AND the applicant is a graduate student, THEN his or her credit is good”.

The semantic heterogeneities between two data resources are represented in a formal way with ontological concepts as well. For example, if one credit card company uses the “age” concept but another one uses “birth_year” to record the applicants’ information, the heterogeneity or mapping between them can also be represented in SWRL:

$$\begin{aligned} \forall x, y \text{ birth_year}(x, y) \rightarrow \text{age}(x, \text{Current_year} - y) \\ \forall x, y \text{ age}(x, y) \rightarrow \text{birth_year}(x, \text{Current_year} - y) \end{aligned}$$

where “Current_year” is a constant of number (e.g., 2011). We notice that RIF [2] is the rule interchange format for the Semantic Web currently under discussion at W3C. As long as RIF becomes a W3C standard, we will change our representation and implementation from SWRL to RIF. The major point of this

paper is that we need an ontology-based rule language to represent data mining knowledge to enable formal reasoning and translation.

4 Knowledge Translation

After we use OWL and SWRL (or RIF) to represent rule-based knowledge and mappings, we can discuss the knowledge translation problem in a formal way.

4.1 Formal Definitions

We first formally define the correctness and completeness of knowledge translation (KT).

Definition 1. Knowledge Translation (KT): Let K_s be the knowledge mined from the source resource with ontology O_s , O_t be the target ontology, and the set of mapping rules between O_s and O_t be Σ . Knowledge translation (KT) is the process to translate K_s represented by O_s to the knowledge K_t represented by O_t according to Σ . We use the symbol \rightsquigarrow_K to indicate the process of knowledge translation.

Since K_s and K_t can be described by ontology languages, they are a set of logical true statements. Σ are also a set of logical true statements as rules.

Definition 2. Correctness of KT: Let K_t be the translated knowledge determined by some algorithm. The K_t is considered as a correct translation from K_s with Σ only if K_t is a semantic consequence of Σ and K_s : $(K_s; \Sigma) \rightsquigarrow_K K_t$ only if $(K_s; \Sigma) \models K_t$, where \models means the semantic consequence (i.e., a logical entailment). It can be implemented using inference as long as the inference is sound: $(K_s; \Sigma) \vdash K_t \Rightarrow (K_s; \Sigma) \models K_t$, where \vdash means an inference.

The above definition for correctness means that if K_s and Σ are true, a correct translation will guarantee that K_t is also true. A correct translation can be implemented by a sound inference. On the contrary, a simple rewriting of rules cannot guarantee to be sound. What kind of inference algorithms can be considered as sound will be further discussed later in this section. Also, if the Σ is 100% true, we expect that K_t has the same accuracy as K_s according to the translated data if it is a correct translation. If the mappings in Σ are not 100% true (i.e., with some uncertainty), the accuracy of K_t depends on the quality of mappings as well. We consider the later a harder problem and further discuss it in Section 6.

Definition 3. Completeness of KT: Let K_t be the translated knowledge generated by some algorithm. K_t is considered as a complete translation from K_s according to Σ only if all statements in K_s can be translated to K_t correctly.

However, a knowledge translation may not be complete if not all concepts in O_s can be mapped to concepts in O_t . It is normally the case for real world data resources. Therefore, we do not focus on getting a fully complete knowledge translation in this research but we will propose a way to handle the uncertainty of mappings in Section 6.

4.2 Asymmetry of Knowledge Translation

Continue with the decision tree rule from a credit card company (e.g., $C1$) data:

$$\forall x, y \text{ Applicant}(x) \wedge \text{age}(x, y) \wedge (y > 30) \wedge \text{grad_student}(x, \text{"Y"}) \rightarrow \text{credit}(x, \text{"Good"})$$

If another credit card company (e.g., $C2$) uses different but semantically related attributes to describe its applicants: “birth_year”, “student” and “credit_ranking”. The mapping rules between corresponding attributes look like:

$$\begin{aligned} \forall x \text{ Applicant}(x) &\leftrightarrow \text{applicant}(x) \\ \forall x, y \text{ birth_year}(x, y) &\rightarrow \text{age}(x, \text{Current_year} - y) \\ \forall x, y \text{ age}(x, y) &\rightarrow \text{birth_year}(x, \text{Current_year} - y) \\ \forall x \text{ grad_student}(x, \text{"Y"}) &\rightarrow \text{student}(x, \text{"Y"}) \\ \forall x, y \text{ credit}(x, y) &\leftrightarrow \text{credit_ranking}(x, y) \end{aligned}$$

A simple rewriting (e.g., “age” \Rightarrow “Current_year” minus “birth_year”, “graduate_student” \Rightarrow “student”, and “credit” \Rightarrow “credit_ranking”) based on the mappings from the source $C1$ to the target $C2$ will get:

$$\forall x, y \text{ Applicant}(x) \wedge \text{student}(x, \text{"Y"}) \wedge \text{birth_year}(x, y) \wedge (\text{Current_year} - y > 30) \rightarrow \text{credit_ranking}(x, \text{"Good"})$$

Careful readers will find that this rule for $C2$ based on the simple rewriting is not necessarily correct nor semantically equivalent to the original rule in $C1$ because that an applicant is a student does not necessarily mean that he or she is a graduate student. However, if $C2$ uses “PhD_student”, “MS_student” and “undergraduate_student” to describe their student applicants, the mappings are:

$$\begin{aligned} \forall x \text{ grad_student}(x, \text{"Y"}) &\leftarrow \text{PhD_student}(x, \text{"Y"}) \\ \forall x \text{ grad_student}(x, \text{"Y"}) &\leftarrow \text{MS_student}(x, \text{"Y"}) \end{aligned}$$

And the translated rules will be:

$$\forall x, y \text{ Applicant}(x) \wedge \text{PhD_student}(x, \text{"Y"}) \wedge \text{birth_year}(x, y) \wedge (\text{Current_year} - y > 30) \rightarrow \text{credit_ranking}(x, \text{"Good"})$$

$$\forall x, y \text{ Applicant}(x) \wedge \text{MS_student}(x, \text{"Y"}) \wedge \text{birth_year}(x, y) \wedge (\text{Current_year} - y > 30) \rightarrow \text{credit_ranking}(x, \text{"Good"})$$

Both translated rules are correct. Therefore, given our formal definition for the correctness of knowledge translation, translation exhibits certain asymmetries that one must be aware of. The translation of condition is different from the translation of conclusion. Assume mined IF-THEN rules are in the form of $L \rightarrow R$ where L is the condition (left side) as the conjunctions of literals and R is the conclusion (right side) as the conjunctions of literals. Consider a general

IF-THEN rule:

$$\forall x_1, x_2 \dots P_1(X) \wedge \dots \wedge P_n(X) \rightarrow Q_1(X) \wedge \dots \wedge Q_m(X)$$

where X is the set of quantified variables $x_1, x_2 \dots$ and constants, the translation of the condition $(P_1(X) \wedge \dots \wedge P_n(X))$ is not the same process as the translation of the conclusion $(Q_1(X) \wedge \dots \wedge Q_m(X))$. We will subscript the symbol \rightsquigarrow with a “L” to indicate the condition translation (\rightsquigarrow_L), and with a “R” to indicate the conclusion translation (\rightsquigarrow_R) in the rest of the paper.

If we transform the IF-THEN rule to the conjunctive normal form (CNF), it becomes:

$$\forall x_1, x_2 \dots \neg P_1(X) \vee \dots \vee \neg P_n(X) \vee (Q_1(X) \wedge \dots \wedge Q_m(X))$$

Instead (not surprisingly), negation ends up involving the same asymmetry as the condition and conclusion translations. Assume that R is an expression which can be derived from Σ and $\neg P$ by inference. Using the deduction theorem in first-order logic and considering that $\neg P \rightarrow R$ is equivalent to $\neg R \rightarrow P$, we know that

$$(\Sigma; \neg P) \vdash R \Leftrightarrow \Sigma \vdash (\neg P \rightarrow R) \Leftrightarrow \Sigma \vdash (\neg R \rightarrow P) \Leftrightarrow (\Sigma; \neg R) \vdash P$$

This gives us a way to translate negations. We can think of P as a “ground condition” ($\theta(P) = P$): Given P , try to find a P' , which satisfies $(\Sigma; P') \vdash P$. But this is just the problem of translating the condition P : $(\Sigma; P) \rightsquigarrow_L P'$.

Therefore, if the condition translation of P is P' , $\neg P'$ can be derived from Σ and $\neg P$ by the conclusion translation and vice versa:

$$\begin{aligned} (\Sigma; P) \rightsquigarrow_L P' &\Rightarrow (\Sigma; \neg P) \rightsquigarrow_R \neg P' \\ (\Sigma; P) \rightsquigarrow_R P' &\Rightarrow (\Sigma; \neg P) \rightsquigarrow_L \neg P' \end{aligned}$$

A similar discussion and a proof on asymmetry of translation are in [11,12].

4.3 Design and Implementation

To address the asymmetry of translation for rule-based knowledge, we extended our open source inference engine, OntoEngine [13], to conduct both condition translation (\rightsquigarrow_L) and conclusion translation (\rightsquigarrow_R) for IF-THEN rules. The basic idea for the translation is:

For each IF-THEN rule in the source, we conduct the condition translation using backward chaining with generalized modus ponens and conduct the conclusion translation using forward chaining with generalized modus ponens. Then we combine the results from backward chaining and forward chaining to a new translated rule in the target. Our contribution is to design a method by combing both backward and forward chaining to address the asymmetry of translation for data mining rules. The knowledge translation algorithm is described in Algorithm 1. The generalized modus ponens is a well known sound

inference procedure. We will illustrate the detail of how our method works with the case studies in Section 5.

OntoEngine is a free downloadable inference engine from SemWebCentral.org. It mainly handles the translation of Semantic Web data (RDF assertions) by forward chaining and the translation of Semantic Web queries by backward chaining. To translate the data mining rules (e.g., decision tree rules and association rules), the major extension we have made in the implementation is to make OntoEngine be able to handle the translation of numeric comparisons (i.e., “>”, “<”, “=”, “>=” and “<=”) as binary predicates and the inference with arithmetic operators (i.e., “+”, “-”, “×” and “÷”) which are not trivial for reasoning.

Algorithm 1. Knowledge Translation (KT)

Input: : Rule $P_1(X) \wedge \dots \wedge P_n(X) \rightarrow Q_1(X) \wedge \dots \wedge Q_m(X)$ in the source ontology.

The mapping rules Σ between the source and target.

Output: : Rule in the target ontology: $P'_1(X) \wedge \dots \wedge P'_u(X) \rightarrow Q'_1(X) \wedge \dots \wedge Q'_v(X)$

$P_T = \text{null}$

for all predicate P_i , $1 \leq i \leq n$ **do**

 Query $P_t = \text{BackwardChaining}(P_i)$

if $P_T == \text{null}$ **then**

$P_T = P_t$

else

if $P_t != \text{null}$ **then**

$P_T = P_T \wedge P_t$

end if

end if

end for

$Q_T = \text{null}$

for all predicate Q_j , $1 \leq j \leq m$ **do**

 Fact $Q_t = \text{ForwardChaining}(Q_j)$

if $Q_T == \text{null}$ **then**

$Q_T = Q_t$

else

if $Q_t != \text{null}$ **then**

$Q_T = Q_T \wedge Q_t$

end if

end if

end for

if $P_T != \text{null}$ and $Q_T != \text{null}$ **then**

$P'_1(X) \wedge \dots \wedge P'_u(X) \Leftarrow P_T$

$Q'_1(X) \wedge \dots \wedge Q'_v(X) \Leftarrow Q_T$

end if

Function BackwardChaining (Query Q)Query $Q_r = \text{null}$ **if** Q 's predicate is in the target ontology **then** $Q_r = Q$ **else****while** $\exists M (\forall x_1 \dots x_k, Pm_1 \wedge \dots \wedge Pm_i \dots \wedge Pm_w \rightarrow \exists z_1 \dots z_l, Qm_1 \wedge \dots \wedge Qm_j \dots \wedge Qm_r)$ in Σ , Q 's predicate is the same as Qm_j in M **do**New Query $Q_N = \text{ModusPonens}(Q, M)$ $Q_r = \text{BackwardChaining}(Q_N)$ **end while****end if**Return Q_r **Function** ForwardChaining (Fact F)Fact $F_r = \text{null}$ **if** F 's predicate is in the target ontology **then** $F_r = F$ **else****while** $\exists M (\forall x_1 \dots x_k, Pm_1 \wedge \dots \wedge Pm_i \dots \wedge Pm_w \rightarrow \exists z_1 \dots z_l, Qm_1 \wedge \dots \wedge Qm_j \dots \wedge Qm_r)$ in Σ , F 's predicate is the same as Pm_i in M **do**New Fact $F_N = \text{ModusPonens}(F, M)$ $F_r = \text{ForwardChaining}(F_N)$ **end while****end if**Return F_r **Function** ModusPonens (Object O , Mapping M)Object $O_r = \text{null}$ **if** O is a Query **then**Query $Q_r = O$; $Substitutions = \{ \}$ **if** Q_r is $Qm_j(?x_j, ?y_j)$ and one predicate in the conclusion of M is $Qm_j(x_j, y_j)$ **then** $Substitutions = Substitutions + \{x_j/?x_j, y_j/?y_j\}$ **end if****if** $Substitutions$ is not empty **then** $O_r = \text{Substitute the variables in the condition (i.e., } Pm_1 \wedge \dots \wedge Pm_i \dots \wedge Pm_w)$
of M .**end if****end if****if** O is a Fact **then**Fact $F_r = O$; $Substitutions = \{ \}$ **if** F_r is $Pm_i(x_i, y_i)$ and one predicate in the condition of M is $Pm_i(?x_i, ?y_i)$ **then** $Substitutions = Substitutions + \{?x_i/x_i, ?y_i/y_i\}$ **end if****if** $Substitutions$ is not empty **then** $O_r = \text{Substitute the variables in the conclusion (i.e., } Qm_1 \wedge \dots \wedge Qm_j \dots \wedge Qm_r)$
of M .**end if****end if**Return O_r

5 Case Studies

5.1 Translation of NBA Classification Rules

In the first case study, which we focused on the translation of decision tree rules, we first extracted data from two popular sports web sites about NBA: the NBA official site¹ and the Yahoo Sports NBA site² and put them into two databases which we called *NBA* and *NBAYahoo*. We then generated two ontologies by transforming relations to OWL classes and attributes to data type properties. Therefore the decision tree rules mined from each database can be represented in OWL and SWRL.

Since both sites collect the data about the same specific domain (i.e., NBA), the data are highly overlapping. However, these two databases use different but semantically related attributes to describe NBA players and teams. For example, in *NBA* the unit of player height is meter but in *NBAYahoo* it is foot. *NBA* uses “position” but *NBAYahoo* uses “playerposition” to describe the positions of players. Therefore the mappings between *NBA* and *NBAYahoo* look like:

- 1 $\forall x \ @NBA:Player(x) \leftrightarrow @NBAYahoo:Player(x)$
- 2 $\forall x, y \ @NBA:height(x, y) \rightarrow @NBAYahoo:height(x, y/0.3048)$
- 3 $\forall x, y \ @NBAYahoo:height(x, y) \rightarrow @NBA:height(x, y * 0.3048)$
- 4 $\forall x, y \ @NBA:weight(x, y) \leftrightarrow @NBAYahoo:weight(x, y)$
- 5 $\forall x, y \ @NBA:position(x, y) \leftrightarrow @NBAYahoo:playerposition(x, y)$

where we use “@NBA:” and “@NBAYahoo:” as namespaces to distinguish the concepts which are from two databases or ontologies but have the same names (e.g., “Player”, “height”, and “weight”).

We ran the C4.5 decision tree learning algorithm (J48 in WEKA [22]) in the *NBA* database and got 11 rules with high accuracy to classify the positions of players based on the “player” table or to classify whether a team will play in playoff based on the “team” table. The overall accuracy of the 11 classification rules is 85.7% (342/399). To show the translation process, we take one example:

$$\forall x, y, z @NBA:Player(x) \wedge Float(y) \wedge Float(z) \wedge @NBA:height(x, y) \wedge @NBA:weight(x, z) \wedge (y < 1.96) \wedge (z \leq 218) \rightarrow @NBA:position(x, 'Guard')$$

This rule means if a player’s height is less than 1.96 meters and weight is less than or equal to 218 pounds, the player is very likely to have the position as “Guard.” The accuracy of this rule is 96.3%. To translate this rule, the first step is conducting the condition translation (\rightsquigarrow_L) as backward chaining from *NBA* to *NBAYahoo*. Note,

$$@NBA:Player(x) \wedge Float(y) \wedge Float(z) \wedge @NBA:height(x, y) \wedge @NBA:weight(x, z) \wedge (y < 1.96) \wedge (z \leq 218)$$

¹ <http://www.nba.com>

² <http://sports.yahoo.com/nba>

is the condition of the rule in the *NBA* ontology. Our inference engine does the backward chaining with mapping rule 1, 3, and 4 and generates the condition of the rule in the *NBAYahoo* ontology:

$$\begin{aligned} & @NBAYahoo:Player(x) \wedge Float(y) \wedge Float(z) \wedge @NBAYahoo:height(x, y) \\ & \wedge @NBAYahoo:weight(x, z) \wedge (y < 6.42) \wedge (z \leq 218) \end{aligned}$$

Similarly, our inference engine conducts the conclusion translation (\rightsquigarrow_R) as forward chaining from *NBA* to *NBAYahoo*. The conclusion of the rule (i.e., $@NBA:position(x, 'Guard')$) is translated to $@NBAYahoo:playerposition(x, 'Guard')$ with mapping rule 5. Then the inference engine combines the results from both the condition translation and conclusion translation with quantifiers:

$$\begin{aligned} \forall x, y, z @NBAYahoo:Player(x) \wedge Float(y) \wedge Float(z) \wedge @NBAYahoo:height(x, y) \wedge \\ @NBAYahoo:weight(x, z) \wedge (y < 6.42) \wedge (z \leq 218) \\ \rightarrow @NBAYahoo:playerposition(x, 'Guard') \end{aligned}$$

This is the translated classification rule in *NBAYahoo*. We tested the accuracy of the translated rule in the *NBAYahoo* data and got a 97.2% accuracy. We ran J48 in WEKA in the *NBAYahoo* data directly, the most similar rule to the above translated rule is

$$\begin{aligned} \forall x, y, z @NBAYahoo:Player(x) \wedge Float(y) \wedge Float(z) \wedge @NBAYahoo:height(x, y) \wedge \\ @NBAYahoo:weight(x, z) \wedge (y < 6.42) \wedge (z \leq 213) \\ \rightarrow @NBAYahoo:playerposition(x, 'Guard') \end{aligned}$$

in which only the splitting point of weight is different (213 vs. 218) and this rule has a 98.5% accuracy. Figure 1 shows the accuracy of all 11 rules mined from *NBA*, 11 translated rules to *NBAYahoo* and 11 most similar rules mined directly from *NBAYahoo*. For most cases translated rules are as accurate as rules mined directly from *NBAYahoo*. Rule 1-7 are about “position” of “player” and rules 8-11 are about “playoff” for “team”.

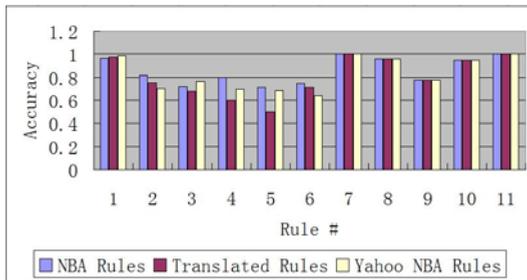


Fig. 1. Accuracy of Classification Rules of the NBA Domain

We tested all 11 translated rules in the *NBAYahoo* data. The overall accuracy is 82.1% (431/525). Compared with the overall accuracy of rules we mined directly from the *NBAYahoo* data as 80.9% (425/525), translated rules have similar overall accuracy as directly mined rules.

We also tested the scalability of our knowledge translation system. For all “player”, “team” and “scores” data tables, we selected different attributes as classification labels and got a large number of classification rules using WEKA. Although many of them are not meaningful or with low accuracy, they are useful for the scalability test. Figure 2 shows that when the number of the rules increases, the processing time of the translation process increases linearly. The testing process was performed on a regular PC with an AMD Athlon Dual-Core Processor 1.90 GHz and 4.00GB memory.

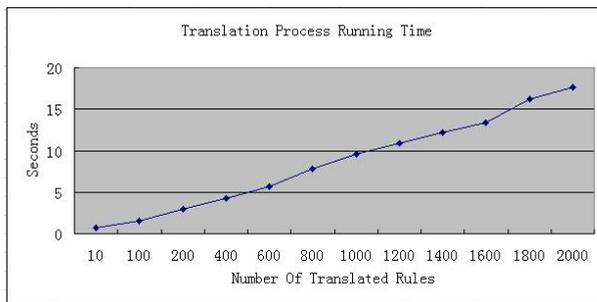


Fig. 2. Scalability of Classification Rule Translation

5.2 Translation of ZFIN and MGI Gene Association Rules

In the second case study, which we focused on the translation of association rules, we obtained the data and schemas from NIH model organisms which support different online gene databases, such as ZFIN³ (the zebrafish gene) and MGI⁴ (the mouse gene). Genetic researchers normally gather knowledge across different species because the genes from different species are potentially related to each other although their data are not overlapping. Comparing the patterns mined from different gene databases are meaningful to domain experts. However, different gene databases use different table names and attributes which make the comparison hard. For both gene databases we tried association rule mining. Then we translated association rules from MGI to the ZFIN schema (ontology) and compared them with the rules mined from ZFIN directly. Domain experts helped us to specify some mappings between MGI and ZFIN so that we could process the translation. The motivation for knowledge translation is that domain experts can compare the association rules from different databases involving shared concepts.

³ <http://www.zfin.org>

⁴ <http://www.informatics.jax.org>

In this case study, we mainly focused on MGI’s “Marker_list” and ZFIN’s “Marker” tables and their attributes. These two tables both describe the information about gene expressions. We got 20 meaningful association rules from the MGI data by using Apriori in WEKA and successfully translated 10 of them to ZFIN. It means 10 of 20 meaningful rules from MGI have corresponding rules as a subset of all rules mined from ZFIN directly. The reason why the rest 10 rules were not translated to ZFIN is that one attribute (i.e, MGI’s “cytogeneticOffset”) has no counterpart in the ZFIN database. This is because that the ZFIN group did not collect these information.

For the 10 rules related to MGI’s “chromosome” and “cM_position” that indicate the position of the marker on the chromosome in mouse genes, it is straightforward to translate them to ZFIN in the similar way as our inference engine does for NBA classification rules. The original data for MGI’s “cM_position” are numbers. We took one preprocessing step to categorize them by selecting some intervals based on its range. For example, one association rule mined from MGI,

$$\forall x @MGI:Marker_list(x) \wedge @MGI:cM_position(x, 16) \rightarrow @MGI:chromosome(x, '1')$$

means if the Marker position on the chromosome is 16, this corresponds to chromosome ‘1’ in the MGI database. To translate this rule, three mappings,

$$\begin{aligned} 6 \quad & \forall x, y @MGI:Market_list(x) \leftrightarrow @ZFIN:Marker(x) \\ 7 \quad & \forall x, y @MGI:cM_position(x, 16) \leftrightarrow @ZFIN:lg_location(x, 20) \\ 8 \quad & \forall x, y @MGI:chromosome(x, '1') \leftrightarrow @ZFIN:or_lg(x, '22') \end{aligned}$$

need to be applied. Note that the ZFIN group uses *or_lg(x, ‘22’)* (linkage groups) rather than chromosome numbers to identify chromosomes. This is because unlike most species, it is hard to readily distinguish one zebrafish chromosome from another using visual techniques. Therefore one linkage group in a zebrafish model includes a set of potential chromosomes and is comparable to a chromosome in a mouse model. The “or” prefix stands for “Oregon” since there are many groups working on linkage groups and the ZFIN group is in Oregon.

The backward chaining for the condition with mapping rule 6 and 7 generates: $@ZFIN:Marker(x) \wedge @ZFIN:lg_location(x, 20)$ and the forward chaining for the conclusion with mapping rule 8 generates: $@ZFIN:lg_location(x, 20)$. Finally the combination generates the translated rule:

$$\forall x @ZFIN:Marker(x) \wedge @ZFIN:lg_location(x, 20) \rightarrow @ZFIN:or_lg(x, '22')$$

The confidence of the original rule in MGI is 67% and the confidence of the translated rule in ZFIN is 56%. Considering the large size of data instances in ZFIN and MGI, although both confidences are not high, this translation and comparison actually shows that there are some similar patterns (associations) with similar confidences in both of the ZFIN and MGI data. From domain

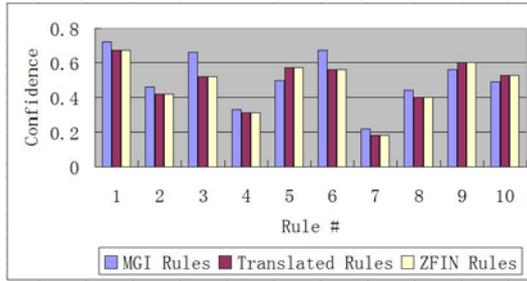


Fig. 3. Confidence of Association Rules of the Gene Domain

experts’ view, this rule also shows there is a marker cluster at a specific location of linkage group 22. Figure 3 shows the confidences of all 10 rules mined from the MGI data, the translated rules from MGI to ZFIN, and the corresponding rules mined from the ZFIN data.

Since the association rule translation process is basically the same as the classification rule translation, we did not perform scalability tests in this case study. Instead, it is more interesting for us to discuss the situation where 10 out of 20 rules are not translatable because MGI’s “cytogeneticOffset” has no counterpart in the ZFIN database. For example, one association rule:

$$\forall x \ @MGI:Marker_List(x) \wedge \ @MGI:Organism_key(x, '1') \wedge \\ \ @MGI:cytogeneticOffset(x, 'p') \rightarrow \ @MGI:chromosome(x, '5')$$

means that if one “Marker” has an organism key as ‘1’ and cytogeneticOffset as ‘p’, the corresponding chromosome will be ‘5’. For MGI’s $_Organism_key(x, '1')$ it should be translated to ZFIN’s $organism(x, 'mouse')$ and MGI’s $chromosome(x, '5')$ should be translated to ZFIN’s $or_lg(x, '24')$. However for MGI’s $cytogeneticOffset(x, y)$ we could not do any translation. Cytogenetics is a set of genetics which describes the structure and function of the chromosome. For the MGI group cytogenetics offset is a very interesting and important attribute while the ZFIN group does not study it because zebrafish genes are not as complex as mouse genes and also too small to collect cytogenetics offset data. But domain experts in ZFIN believe rules with this attribute are interesting and meaningful. The suggestion was to create a “cytogeneticOffset” attribute in ZFIN and the rest 10 rules can be translated successfully. For example, one translated rule is:

$$\forall x \ @ZFIN:Marker(x) \wedge \ @ZFIN:organism(x, 'mouse') \wedge \\ \ @ZFIN:cytogeneticOffset(x, 'p') \rightarrow \ @ZFIN:or_lg(x, '24')$$

6 Discussion and Future Work

In our work, we have specified the mappings among data resources manually. Although it is a one-time job compared with potential applications for translating

large number of various data mining rules, mapping specification is still time consuming. Some automatic or semi-automatic mapping discovery tools, such as schema mapping [5,18] and ontology mapping tools [21], will be helpful. However, there must be some uncertainty with the automatically discovered mappings. It also happens for the mappings specified by human experts, because sometime it is hard to say what exact mappings are among attributes.

To handle mappings with uncertainty, one promising way is to extend Semantic Web ontologies with Markov logic [9], which combines first-order logic with Markov random fields, to represent knowledge and mappings as Markov logic networks (MLNs). The knowledge translation with uncertain mappings can be a process using both logic inference and probabilistic inference.

We also plan to apply knowledge translation algorithms for distributed data mining (DDM) systems in a client-server model. The clients will be data analysts and a DDM server will connect to local data resources. The data mining tasks will run on local resources. The output from local data resources is the mined knowledge based on local ontologies. Given the mappings between local data resources and the user site, the system will apply appropriate knowledge translation algorithms to first translate the knowledge to the user ontology, then the knowledge from multiple resources can be combined as if in the homogeneous scenario (i.e., the same user ontology).

7 Conclusions

Major contributions of this research to data mining and the Semantic Web are:

- It is novel to research how to translate the mined knowledge from one data resource to another semantically heterogeneous one. This work can be applied to knowledge transfer and potentially to distributed data mining in the semantically heterogeneous scenario.
- The formal representation of mined knowledge leverages the ideas of the Semantic Web to make the knowledge computer “understandable” and “translatable”. It is a key step to make the mined knowledge sharable among software agents.
- The general nature of our approach makes it applicable to any domain, especially to biomedical sciences, where large amounts of data are already publicly available from different labs but are semantically heterogeneous.

In our future work, we need to consider the uncertainty of mappings. We also plan to extend our knowledge translation methods to distributed data mining.

Acknowledgment. We thank Xiang Shao, Sridhar Ramachandran, and Tom Conlin in the ZFIN group for providing domain knowledge on genetic data, database mappings, and valuable comments.

References

1. OWL Web Ontology Language, <http://www.w3.org/TR/owl-ref/>
2. Rule Interchange Format (RIF), <http://www.w3.org/2005/rules/>
3. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, <http://www.w3.org/Submission/SWRL/>
4. XLMiner: Data Mining in Excel, <http://www.resample.com/xlminer/index.shtml>
5. An, Y., Borgida, A., Miller, R.J., Mylopoulos, J.: A semantic approach to discovering schema mapping expressions. In: ICDE, pp. 206–215 (2007)
6. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American* 284(5) (May 2001)
7. Bruijn, J.D., Polleres, A.: Towards an Ontology Mapping Specification Language for the Semantic Web. Technical report, DERI (June 2004)
8. Caragea, D., Zhang, J., Bao, J., Pathak, J., Honavar, V.G.: Algorithms and software for collaborative discovery from autonomous, semantically heterogeneous, distributed information sources. In: Jain, S., Simon, H.U., Tomita, E. (eds.) ALT 2005. LNCS (LNAI), vol. 3734, pp. 13–44. Springer, Heidelberg (2005)
9. Domingos, P., Lowd, D.: *Markov Logic: An Interface Layer for AI*. Morgan & Claypool, CA (2009)
10. Dou, D., Frishkoff, G., Rong, J., Frank, R., Malony, A., Tucker, D.: Development of NeuroElectroMagnetic Ontologies (NEMO): A Framework for Mining Brainwave Ontologies.. In: KDD, pp. 270–279 (2007)
11. Dou, D., McDermott, D.: Deriving Axioms Across Ontologies. In: AAMAS, pp. 952–954 (2006)
12. Dou, D., McDermott, D.: Towards theory translation. In: Baldoni, M., Endriss, U. (eds.) DALT 2006. LNCS (LNAI), vol. 4327, pp. 16–28. Springer, Heidelberg (2006)
13. Dou, D., McDermott, D.V., Qi, P.: Ontology Translation on the Semantic Web. *Journal on Data Semantics* 2, 35–57 (2004)
14. Eaton, E.: Multi-resolution learning for knowledge transfer. In: AAAI (2006)
15. Gao, J., Fan, W., Jiang, J., Han, J.: Knowledge transfer via multiple model local structure mapping. In: KDD, pp. 283–291 (2008)
16. Gupta, R., Ratinov, L.-A.: Text categorization with knowledge transfer from heterogeneous data sources. In: AAAI, pp. 842–847 (2008)
17. Liu, H., Dou, D.: An Exploration of Understanding Heterogeneity through Data Mining. In: Proceedings of KDD 2008 Workshop on Mining Multiple Information Sources, pp. 18–25 (2008)
18. Miller, R.J., Hernández, M.A., Haas, L.M., Yan, L.-L., Ho, C.T.H., Fagin, R., Popa, L.: The clio project: Managing heterogeneity. *SIGMOD Record* 30(1), 78–83 (2001)
19. Park, B.-H., Kargupta, H.: Distributed data mining: Algorithms, systems, and applications. In: Ye, N. (ed.) *Data Mining Handbook* (2002)
20. Provost, F.J., Buchanan, B.G.: Inductive policy: The pragmatics of bias selection. *Machine Learning* 20(1-2), 35–61 (1995)
21. Qin, H., Dou, D., LePendu, P.: Discovering Executable Semantic Mappings Between Ontologies. In: ODBASE, pp. 832–849 (2007)
22. Witten, I.H., Frank, E.: *Data Mining: Practical machine learning tools and techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)