A Probabilistic Approach to Knowledge Translation

Shangpu Jiang, Daniel Lowd, Dejing Dou
Computer and Information Science
University of Oregon, USA
{shangpu,lowd,dou}@cs.uoregon.edu

Abstract

In this paper, we focus on a novel knowledge reuse scenario where the knowledge in the source schema needs to be translated to a semantically heterogeneous target schema. We refer to this task as "knowledge translation" (KT). Unlike data translation and transfer learning, KT does not require any data from the source or target schema. We adopt a *probabilistic approach* to KT by representing the knowledge in the source schema, the mapping between the source and target schemas, and the resulting knowledge in the target schema all as probability distributions, specially using *Markov random fields* and *Markov logic networks*. Given the source knowledge and mappings, we use standard learning and inference algorithms for probabilistic graphical models to find an explicit probability distribution in the target schema that minimizes the Kullback-Leibler divergence from the implicit distribution. This gives us a compact probabilistic model that represents knowledge from the source schema as well as possible, respecting the uncertainty in both the source knowledge and the mapping. In experiments on both propositional and relational domains, we find that the knowledge obtained by KT is comparable to other approaches that require data, demonstrating that knowledge can be reused without data.

Introduction

Knowledge acquisition is a critical process for building predictive or descriptive models for many applications. When domain expertise is available, knowledge can be constructed manually. When enough high-quality data is available, knowledge can be constructed automatically using data mining or machine learning tools. Both approaches can be difficult and expensive, so we would prefer to reuse or transfer knowledge from one application or system to another whenever possible. However, different applications or systems often have different semantics, which makes knowledge reuse or transfer a non-trivial task.

As a motivating example, suppose a new credit card company without historical data wants to use the classification model mined by a partner credit card company to determine whether the applicants of the new company are qualified or not. Since the two companies may use different schemas to store their applicants' data (e.g., in one schema, we have annual income recorded as a numerical attribute, while in the other, we have salary as an attribute with discretized ranges), we cannot simply reuse the old classifier. Due to privacy and scalability concerns, we cannot transfer the collaborative company's data to the new schema, either. Therefore, we want to *translate* the classification model itself to the new schema, *without using any data*.

In this paper, we propose *knowledge translation* (KT) as a novel solution to translate knowledge across conceptually similar but semantically heterogeneous schemas or ontologies. For convenience, we refer to them generically as "schemas." As shown in the previous example, KT is useful in situations where data translation/transfer is problematic due to privacy or scalability concerns.

We formally define knowledge translation as the task of converting knowledge $K_{\mathcal{S}}$ in source schema $\mathcal{S}$ to equivalent knowledge $K_{\mathcal{T}}$ in target schema $\mathcal{T}$, where the correspondence between the schemas is given by some mapping $\mathcal{M}_{\mathcal{S},\mathcal{T}}$. In general, one schema may have concepts that are more general or specific than the other, so an exact translation may not exist. We will therefore attempt to find the *best* translation, acknowledging that the best translation may still be a lossy approximation of the source knowledge.

We adopt a *probabilistic approach* to knowledge translation, in which the knowledge in the source schema, the mapping between the source and target schemas, and the resulting knowledge in the target schema are all represented as probability distributions. This gives us a consistent mathematical framework for handling uncertainty at every step in the process. This uncertainty is clearly necessary when the source knowledge is probabilistic, but it is also necessary when there is no exact mapping between the schemas, or when the correct mapping is uncertain. We propose to represent these probability distributions using *Markov random fields*, for propositional (non-relational) domains, and *Markov logic networks*, for relational domains. Given probability distributions for both the source knowledge and the schema mapping, we can combine them to define an implicit probability distribution in the target schema. Our goal is to find an explicit probability distribution in the target schema that is close to this implicit distribution in terms of the Kullback-Leibler divergence.

Our main contributions are:

- We formally define the problem of knowledge translation (KT), which allows knowledge to be reused for heterogeneous schemas when data is unavailable.

- We propose a novel *probabilistic* approach for KT.
- We implement an experimental KT system and evaluate it on two real datasets. We compare our data-free KT approach to baselines that use data from the source or target schema and show that we can obtain comparable accuracy without data.

The paper is organized as follows. We first summarize related work, such as semantic integration, distributed data mining, and transfer learning, and discuss their connections and distinctions with KT. We then show how Markov random fields and Markov logic networks can represent knowledge and mappings with uncertainty. Next, we present a variant of the MRF/Markov logic learning algorithm to solve the problem of knowledge translation. We then run experiments on synthetic and real datasets. Finally, we conclude and outline future work.

Related Work

In this section, we compare the task of knowledge translation with some related work (See Table 1).

Semantic Integration Data integration and exchange (e.g., (Lenzerini 2002)) are the most studied areas in semantic integration. The main task of data integration and exchange is to answer queries posed in terms of the global schema, given source databases. The standard semantics of global query answering is to return the tuples in every possible database that is consistent with the global schema constraints and the mapping, i.e., the set of *certain answers*.

A main difference between data integration/exchange and knowledge translation (KT) is that KT has probabilistic semantics for the translation process, that is, it defines a distribution of possible worlds in the target schema, instead of focusing only on the tuples that are in all the possible worlds (i.e., certain answers).

Distributed Data Mining Efforts in distributed data mining (DDM) (see surveys in (Park and Kargupta 2002; Caragea et al. 2005)) have made considerable progress in mining distributed data resources without putting data in a centralized location. (Caragea et al. 2005) proposes a general DDM framework with two components: one sends statistical queries to local data sources, and the other uses the returned statistics to revise the current partial hypothesis and generate further queries.

Heterogeneous DDM (Caragea et al. 2005) also handles the semantic heterogeneity between the global and local schemas, in particular those containing attributes with different granularities called Attribute Value Taxonomy (AVT). Heterogeneous DDM requires local data resources and their mappings to the global schema to translate the statistics of queries. However, KT does not require data from either the source or the target. Instead, KT uses mappings to translate the generated/mined knowledge from the source directly.

Transfer Learning Transfer learning (TL) has been a successful approach to knowledge reuse (Pan and Yang 2010). In traditional machine learning, only one domain and one task is involved. When the amount of data is limited, it is desirable to use data from related domains or tasks. As long as the source and target data share some similarity (e.g., in the distribution or underlying feature representation), the knowledge obtained from the source data can be used as a "prior" for the target task.

Early transfer learning work focuses on the homogeneous case in which the source and target domain have identical attributes. Recently, many other scenarios of transfer learning have been studied, including heterogeneous transfer learning (Yang et al. 2009), relational transfer learning (Mihalkova, Huynh, and Mooney 2007; Davis and Domingos 2009), and network transfer learning (Fang et al. 2015; Ye et al. 2013). Some of these scenarios have similar settings as knowledge translation. Heterogeneous transfer learning also deals with different representations of the data. While it uses an implicit mapping of two feature spaces (e.g., texts and images through the tags on Flickr), KT uses an explicit mapping via FOL formulas. Relational transfer learning also involves relational domains and relational knowledge. While it deals with two analogous domains (e.g., in the movie and university domains, directors correspond to professors), KT focuses on a single domain with merely different representations. Moreover, relational transfer learning only handles deterministic one-to-one matchings which can be inferred with both the source and target data, while KT does not use any target data and relies on the provided explicit FOL mapping.

Deductive Knowledge Translation Deductive knowledge translation (Dou, Qin, and Liu 2011) essentially tries to solve the same problem, but it only considers deterministic knowledge and mappings. Our new KT work can handle knowledge and mappings with uncertainty, which is more general than the deterministic scenario deductive knowledge translation can handle.

Table 1: Comparisons between KT and related work. We consider three aspects of a task: whether data is available, what kind of knowledge patterns are supported, and what kind of mapping is used.

	Data avail.	Knowledge	Mapping
Data integration	Source data	Query results	GLAV
Hetero. DDM	Source data	Propositional	AVT
Hetero. TL	Source/target	any	Implicit
Relational TL	Target data	SRL models	Matching
Deductive KT	No data	FOL	FOL
KT	No data	SRL models	SRL models

Probabilistic Representations of Knowledge and Mappings

To translate knowledge from one schema to another, we must have a representation of the knowledge and the mappings between the two schemas. In many cases, knowledge and mappings are uncertain. For example, the mined source knowledge could be a probabilistic model, such as a Bayesian network. Mappings between two schemas may also be uncertain, either because a perfect alignment of the concepts does not exist, or because there is uncertainty about which alignment is the best. Therefore, we propose a *probabilistic* approach to knowledge translation.

Probabilistic Representations

A log-linear model is a compact way to represent a positive probability distribution $p(\boldsymbol{X})$ over a set of random variables $\boldsymbol{X} = \{X_1, X_2, \ldots, X_N\}$. In a log-linear model, the probability of any configuration $\boldsymbol{x}$ is defined as

$$p(\boldsymbol{X} = \boldsymbol{x}) = \frac{1}{Z} \exp\left(\theta^T \phi(\boldsymbol{x})\right),$$

where $\phi(\boldsymbol{x})$ is a vector-valued feature function, θ is a real-valued feature vector, and Z is a normalization constant. Probabilistic graphical models, such as Bayesian networks (Pearl 1988) and Markov random fields (MRFs (Kindermann, Snell, and others 1980)), can be represented as log-linear models.

The area of statistical relational learning (SRL) (Getoor and Taskar 2007) explores representation, learning, and inference of probabilistic models in relational domains. One of the most powerful statistical relational representations to date is Markov logic (Domingos and Lowd 2009). A Markov logic network (MLN) consists of a set of weighted formulas in first-order logic, $\{(f_i, w_i)\}$. Together with a finite set of constants, an MLN defines a probability distribution over possible worlds (complete assignments of truth values to atoms) by using the number of true groundings of each formula as a feature in a log-linear model:

$$p(\boldsymbol{X} = \boldsymbol{x}) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(\boldsymbol{x})\right)$$

where $n_i(\boldsymbol{x})$ is the number of times the ith formula is satisfied by the possible world $\boldsymbol{x}$. Z is a normalization constant.

Representation of Knowledge

Our approach to knowledge translation requires that the source and target knowledge are probability distributions represented as log-linear models. In some cases, the source knowledge mined from the data may already be represented as a log-linear model, such as a Bayesian network used for fault diagnosis or Markov logic network modeling homophily in a social network. In other cases, we will need to convert the knowledge into this representation.

For mined knowledge represented as rules, including association rules, rule sets, and decision trees (which can be viewed as a special case of rule sets), we can construct a feature for each rule, with a weight corresponding to the confidence or probability of the rule. The rule weight has a closed-form solution based on the log odds that the rule is correct:

$$w_i = \log \frac{p(f_i)}{1 - p(f_i)} - \log \frac{u(f_i)}{1 - u(f_i)}$$

where $p(f_i)$ is the probability or confidence of the ith rule or formula and $u(f_i)$ is its probability under a uniform distribution. Relational rules in an ontology can similarly be converted to a Markov logic network by attaching weights representing their relative strengths or confidences.

For linear classifiers, such as linear support vector machines or perceptrons, we can substitute logistic regression, a probabilistic linear classifier.

In some cases, the knowledge we wish to translate takes the form of a conditional probability distribution, $p(\boldsymbol{Y}|\boldsymbol{X})$, or a predictive model that can be converted to a conditional probability distribution. This includes decision trees, neural networks, and other classifiers used in data mining and machine learning. The method we propose will rely on a full joint probability distribution over all variables. We can convert a conditional distribution into a joint distribution by assuming some prior distribution over the evidence, $p(\boldsymbol{X})$, such as a uniform distribution.

Representation of Mappings

The relationships between heterogeneous schemas can be represented as a *mapping*. We use probabilistic models to represent mappings. Consistent with the probabilistic representation of knowledge in a database schema, the attributes are considered as random variables for non-relational domains, and the attributes or relations are considered as first-order random variables for relational domains. Let us denote the variables in the source as $\boldsymbol{X} = \{X_1, ..., X_N\}$ and those in the target as $\boldsymbol{X}' = \{X'_1, ..., X'_M\}$. A mapping is the conditional distribution $p(\boldsymbol{X}'|\boldsymbol{X})$.

In practice, a mapping is often decomposable to a set of roughly independent source-to-target correspondences

$$\{p(\boldsymbol{C}'_i|\boldsymbol{C}_i), i = 1, ..., I\}$$

where $\boldsymbol{C}_i \subset \boldsymbol{X}$ and $\boldsymbol{C}'_i \subset \boldsymbol{X}'$ are subsets of source and target variables respectively. For the credit card company example, a mapping between the two schemas may include the correspondences of "age" and "age," "salary" and "annual income," etc. Formally, we say correspondences are *independent* when each group of target variables is conditionally independent from all other variables given the matching source variables: $\boldsymbol{C}'_i \perp \boldsymbol{X}, \boldsymbol{X}' \setminus \boldsymbol{C}'_i | \boldsymbol{C}_i$. If the correspondences are independent or approximately independent, then we can approximate the global mapping with the local correspondences:

$$p(\boldsymbol{X}'|\boldsymbol{X}) \approx \prod_i p(\boldsymbol{C}'_i|\boldsymbol{C}_i)$$

We encode each correspondence as weighted propositional or first-order logical formulas, where the weight can be estimated with the log-odds. $p(\boldsymbol{X}'|\boldsymbol{X})$ is then simply a log-linear model with these formulas as features. For example, we define a probabilistic source-to-target correspondence as $q_{\mathcal{S}} \rightarrow_p q_{\mathcal{T}}$, where $q_{\mathcal{S}}$ and $q_{\mathcal{T}}$ are queries (i.e., logical formulas) of source and target schemas or ontologies, and $\rightarrow_p$ has probabilistic semantics:

$$\Pr(q_{\mathcal{T}}|q_{\mathcal{S}}) = p$$

Example 1 (Class correspondence)**.** If x is a graduate student, then x is a student and older than 24 with probability 0.9, and vice versa.

$$\texttt{Grad}(x) \rightarrow_{0.9} \texttt{Student}(x) \wedge \texttt{Age}(x,y) \wedge (y \geq 24)$$
$$\texttt{Grad}(x) \leftarrow_{0.9} \texttt{Student}(x) \wedge \texttt{Age}(x,y) \wedge (y \geq 24)$$

This can be converted to

$$2.2 \quad \texttt{Grad}(x) \rightarrow (\texttt{Student}(x) \wedge \texttt{Age}(x,y) \wedge (y \geq 24))$$
$$2.2 \quad \texttt{Grad}(x) \leftarrow (\texttt{Student}(x) \wedge \texttt{Age}(x,y) \wedge (y \geq 24))$$

Dong et al. (2007; 2009) proposed *probabilistic schema mappings* to handle the *uncertainty* in mappings, which is similar to our method. They define probabilistic mapping as a set of mapping and probability pairs $\{\sigma_i, \Pr(\sigma_i)\}, i = 1, \cdots, l$, where $\sum_{i=1}^{l} Pr(\sigma_i) = 1$. In the by-tuple semantics they defined, different mappings can be applied to different tuples of a table, which is equavalent to the semantics of our probabilistic mapping representation. Our representation further extends their method from one-to-one mappings to arbitrary FOL mappings.

Knowledge Translation

In this section, we formalize the task of knowledge translation (KT) and propose a solution to this task. In KT, we are given the source knowledge represented as a probabilistic model $p(\boldsymbol{X}) = p(X_1, \ldots, X_n)$ and a probabilistic mapping $p(\boldsymbol{X}'|\boldsymbol{X})$. The probabilistic model in the target schema can be computed as

$$p(\boldsymbol{X}') = \sum_{\boldsymbol{X}} p(\boldsymbol{X})p(\boldsymbol{X}'|\boldsymbol{X}) = \sum_{\boldsymbol{X}} p(\boldsymbol{X}) \prod_i p(\boldsymbol{C}'_i|\boldsymbol{C}_i) \tag{1}$$

The goal of KT is to approximate this distribution with a *compact* probabilistic model $q(\boldsymbol{X}')$ in the target schema without using any source variables as latent variables. This requirement helps make the knowledge more efficient to use and easier to understand.

A straight-forward objective is to minimize the Kullback-Leibler divergence

$$\begin{aligned} q^* &= \arg\min_q D_{\mathrm{KL}} \left[p(\boldsymbol{X}') \| q(\boldsymbol{X}') \right] \\ &= \arg\min_q -E_p[\log q(\boldsymbol{X}')] \end{aligned} \tag{2}$$

Parameter Learning The objective described in Equation 2 is typically intractable to compute exactly, due to the expectation over $p(\boldsymbol{X}')$, but we can approximate it using samples. To generate a sample from $p(\boldsymbol{X}')$, we first generate a sample from $p(\boldsymbol{X})$ and then generate a sample of $\boldsymbol{X}'$ from $p(\boldsymbol{X}'|\boldsymbol{X})$ conditioned on the source sample. In a relational domain (with Markov logic or other statistical relational models), each sample instance is a database, and we need to first decide the number of constants and create a set of ground variables with these constants.

After replacing the expectation in Equation 2 with a sum over samples, the objective is simply the negative log-likelihood of the samples S: $\sum_{\boldsymbol{x}' \in S} - \log q(\boldsymbol{X}')$. If q is represented as a log-linear model, then its parameters can be optimized using standard weight learning algorithms (e.g., (Lowd and Domingos 2007)).

Structure Learning The structure of the target knowledge can also be learned from samples via standard structure learning algorithms for Markov random fields or Markov logic networks. An alternative approach is to use heuristics to generate the structure first. For deterministic one-to-one correspondences, the independences in the target schema ($p(\boldsymbol{X}')$) are the same as those in the source schema ($p(\boldsymbol{X})$) up to renaming. If the correspondences are non-deterministic, the summation in Equation 1 may lead to a distribution $p(\boldsymbol{X}')$ with few or no independences. Representing this structure exactly with no latent variables would require an extremely complex model with large cliques. Nonetheless, in realistic scenarios, the correspondences in a mapping are usually deterministic or nearly deterministic. Therefore, it is reasonable to treat them as deterministic while inferring the target structure. In this way we trade off between the complexity and accuracy of the target knowledge.

We present the pseudocode of our heuristic structure translation in Algorithm 1. For Markov random fields, the structure can be described as a set of cliques. For Markov logic, we use first-order cliques instead of formulas as the source structure, so that it is consistent with the propositional case. The first step (Line 1–8) is to remove the variables that do not have a correspondence in the target schema. In standard variable elimination (Koller and Friedman 2009; Poole 2003), we can remove a variable from a network structure by merging all of its neighboring cliques into a new clique. (Since we are only concerned with structure, we do not need to compute the parameters of the modified network.) However, this procedure may create very large cliques, especially in Markov logic in the relational domains. Therefore, we approximate it by only merging two cliques at a time. For the relational case, the merging involves a first-order unification operation (Russell and Norvig 2003; Poole 2003). When multiple most general unifiers exist, we simply include all the resulting new cliques. In the second step (Line 9–15), we replace each variable with the corresponding variable in the target schema. This also involves first-order unification in the relational case. If there are many-to-many correspondences, we may generate multiple target cliques from one source clique.

Example 2. Given the source Markov logic network:

$$\texttt{Grad}(x) \rightarrow \texttt{AgeOver25}(x)$$
$$\texttt{AgeOver25}(x) \rightarrow \texttt{GoodCredit}(x)$$

and the mapping:

$$2.2 \quad \texttt{Grad}(x) \vee \texttt{Undergrad}(x) \leftrightarrow \texttt{Student}(x)$$
$$3.0 \quad \texttt{GoodCredit}(x) \leftrightarrow \texttt{HighCreditScore}(x)$$

We first eliminate $\texttt{AgeOver25}(x)$ from the source structure because it does not occur in the mapping, and we get a new (first-order) clique:

$$\{\texttt{Grad}(x), \texttt{GoodCredit}(x)\}$$

Algorithm 1 Structure Translation (MRFs or MLNs)

```
Input: The source schema $\mathcal{S}$, source structure (propositional or first-order cliques) $\Phi = \{\phi_i\}$, and mapping $\mathcal{M}$.
Output: The target structure $\Phi'_{\mathcal{M}}$.
for each variable (or first-order predicate) $P \in \mathcal{S}$ that does not appear in $\mathcal{M}$ do
    $\Phi_P \leftarrow$ the set of cliques containing $P$
    Remove $\Phi_P$ from $\Phi$
    for each pair of cliques in $\Phi_P$ do
        Merge the two cliques and remove $P$
        Insert the resulting clique back to $\Phi$
    end for
end for
for each clique $\phi \in \Phi$ do
    for each variable $P$ (or first-order atom) in $\phi$ do
        Let $P'_{\mathcal{M}}$ be all possible correspondences of $P$
    end for
    Let $\phi'_{\mathcal{M}}$ denote the correspondences of $\phi$: $\phi'_{\mathcal{M}} \leftarrow$ Cartesian product of all $P'_{\mathcal{M}}$
    Add $\phi'_{\mathcal{M}}$ to $\Phi'_{\mathcal{M}}$
end for
```

Then we translate the clique based on the mapping, which gives:

$$\{\texttt{Student}(x), \texttt{HighCreditScore}(x)\}$$

Experiments

To evaluate our methods, we created two knowledge translation tasks: one on a non-relational domain (NBA) and one on a relational domain (University). In each knowledge translation task, we have 2 different database schemas as the source and target schemas and a dataset for each schema. The input of a knowledge translation system is the source knowledge and the mapping between the source and target schema. The output of a knowledge translation system is the target knowledge (i.e., a probabilistic model in the target schema).

We obtained the source knowledge (i.e., a probabilistic model in the source) by performing standard learning algorithms on the source datasets, and created the probabilistic schema mappings manually. Our approach can potentially use automatically discovered mappings (e.g., (Rahm and Bernstein 2001)) as well, but we use manually created mappings in the experiments for two reasons: first, our method strongly relies on the quality of the mapping, so we want to use more accurate mappings for a quantitative analysis of the method itself; second, we use schemas with plenty of semantic heterogeneity to make the translation problem non-trivial, which is a difficult scenario for automatic tools and the quality of discovered mappings is not guaranteed.

Methods and Baselines

We evaluate three different versions of our proposed probabilistic knowledge translation approach described in the previous section. All of them use the source knowledge and probabilistic mapping to generate a sampled approximation of the distribution in the target schema, and all of them use these samples to learn an explicit distribution in the target schema. The difference between them is their approach to knowledge structure. **LS-**K_S ("learned structure") learns the structure directly from the samples, which is the most flexible approach. **TS-**K_S ("translated structure") uses a heuristic translation of the structure from the source knowledge base. **ES-**K_S ("empty structure") is a baseline in which the target knowledge base is limited to a marginal distribution.

We also compare to several baselines that make use of additional data. When there is data D_S in the source schema, we can use the probabilistic mapping to translate it to the target schema and learn models from the translated source data. **LS-**D_S and **MS-**D_S learn models from translated source data, using learned and manually specified structures, respectively. When there is data D_T in the target schema, we can learn from this data directly. **LS-**D_T and **MS-**D_T learn models from target data with learned and manually specified structures respectively. These methods represent an unrealistic "best case" since they use data that is typically unavailable in knowledge translation tasks.

We evaluate our knowledge translation methods according to two criteria: the pseudo-log-likelihood (PLL) on the held-out *target data*, and PLL on the held-out *translated source data*. The advantage of the second measure is that it controls for differences between the source and target distributions. For relational domains, we use weighted pseudo-log-likelihood (WPLL), where for each predicate r, the PLL of each of its groundings is weighted by the $c_r = 1/g_r$, where g_r is the number of its groundings.

Non-Relational Domain (NBA)

We collected information on basketball players in the National Basketball Association (NBA) from two websites, the NBA official website `nba` (as the source schema) and the Yahoo NBA website `yahoo` (as the target schema). The schemas of these two datasets both have the name, height, weight, position and team of each player. In these schemas, the values of position have a different granularity. Also, in `nba`, we discretize height and weight into 5 equal-width ranges. In `yahoo`, we discretize them into 5 equal-frequency ranges (in order to make the mapping more challenging). The correspondences of these attributes are originally unit conversion formulas, e.g., $h' = h \times 39.37$. After we discretize these attributes, we calculate the correspondence distribution of the ranges by making a simple assumption that each value range is uniformly distributed, e.g.,

$$p(h' \in (73.5, 76.5] | h \in (1.858, 1.966]) = 0.706$$

We used the Libra Toolkit[1](Lowd and Rooshenas 2015) for creating the source knowledge and for performing the learning and inference subroutines required by the different knowledge translation approaches. We first left out 1/5 of the data instances in the source and target dataset as the testing sets. For the remaining source dataset, we used the decision tree structure learning (DTSL) (Lowd and Davis 2014) to learn a Markov random field as the source knowledge.

[1] `http://libra.cs.uoregon.edu/`

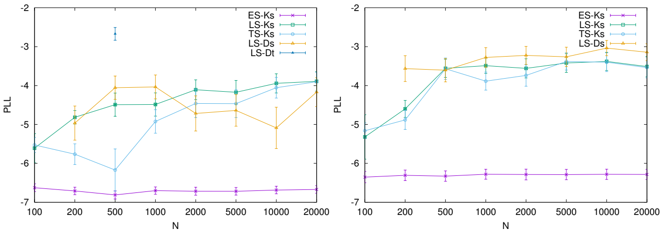

Figure 1: PLL for KT methods and baselines on target data (left) and translated source data (right) in the NBA domain.

We used standard 4-fold cross-validation to determine the parameters of the learning algorithm. The parameters include κ, prior, and mincount for decision tree learning, and l_2 for weight learning. The final source MRF have about 100 conjunctive features, with the maximum length of 7.

We use Gibbs sampling for the sampling algorithm in the knowledge translation approaches. For LS-K_S and TS-K_S, we draw N samples from the source knowledge probability distribution. We then use the probabilistic mapping to draw 1 target sample for each source sample. For LS-D_S, suppose we have N_S instances in the source dataset. We use the probabilistic mapping to draw N/N_S target samples for each source instance, such that the total number of target instances is also N.

LS-K_S and TS-K_S both perform weight learning with an l_2 prior. For structure translation with TS-K_S, we only translate features for which the absolute value of the weight is greater than a threshold θ. These two parameters are tuned with cross-validation over a partition of the samples.

Figures 1 shows learning curves comparing our methods to the baselines. We see that translated knowledge (LS-K_S and TS-K_S) is as accurate as knowledge learned from translated source data (LS-D_S) on both the target data and the translated source data. This confirms that *KT can be as accurate as data translation*, but with the advantage of not requiring any data. We do not see a large difference between learning the structure (LS-K_S) and heuristically translating the structure (TS-K_S). As expected, the model learned directly on the target data (LS-D_T) has the best PLL on the target data, since it observes the target distribution directly.

Relational Domain (University)

We use the UW-CSE dataset[2] and the UO dataset which we collected from the Department of Computer Science of the University of Oregon. The UW-CSE dataset was introduced by Richardson and Domingos (Richardson and Domingos 2006) and is widely used in statistical relational learning research. In this University domain, we have concepts such as persons, courses, and publications; attributes such as PhD student stage and course level; and relations such as advise, teach, and author. The schemas of the two databases differ in their granularities of concepts and attribute values. For example, UW-CSE graduate courses are marked as level 500, while UO has both graduate courses at level 600 and combined undergraduate/graduate courses at level 4/500. Our methods in this relational domain are similar to those in the non-relational domain. We use Alchemy[3] for learning and inference in Markov logic networks. We obtain the source knowledge by manually creating formulas in the source schema and then using the source data to learn the weights. We have about 100 formulas and most of them are clauses with 2 literals.

Table 2: Evaluation on the target dataset (left) and translated source dataset (right) for the university domain. N/A means that structure learning takes more than 1 day.

Method	WPLL on target			WPLL on source		
# Samples	1	2	5	1	2	5
ES-K_S	-3.77	-3.76	-3.83	-3.54	-3.44	-3.39
LS-K_S	-12.07	-3.82	-3.48	-9.19	-3.72	-1.51
TS-K_S	-2.51	-2.80	-1.79	-2.05	-2.10	-0.97
LS-D_S	-3.70	-3.01	N/A	-1.23	-1.23	N/A
MS-D_S	-1.94	-1.91	-1.76	-1.22	-0.93	-0.61
LS-D_T	-1.33					
MS-D_T	-1.18					

We use MC-SAT (Poon and Domingos 2006) as the sampling algorithm for these experiments. Since the behavior of a Markov logic network is highly sensitive to the number of constants, we want to keep the number of constants similar to the original dataset from which the model is learned. We set the number of constants of each type to be the average number over all training databases, multiplied by a scalar $\frac{1}{2}$ for more efficient inference. For methods based on K_S, we draw N samples from the source distribution and 1 target sample from each source sample and the mapping. For methods based on D_S, we draw N samples based on the mapping. Here N does not have to be large, because each sample instance of a relational domain is itself a database. We set N to 1, 2 and 5 in our experiments. We set the l_2 prior for weight learning to 10, based on cross-validation over samples.

The results are shown in Table 2. In general, learning MLN structure (LS-K_S and LS-D_S) did not work as well as their counterparts with translated or manually specified structures (TS-K_S and MS-D_S). From a single sample, the translated source data and manually specified structure (MS-D_S) were more effective than knowledge translation with translated structure (TS-K_S). However, as we increase the number of samples, the performance of TS-K_S improves substantially. With 5 samples, the performance of TS-K_S becomes competitive with that of MS-D_S, again demonstrating that knowledge translation can achieve comparable results to data translation but without data. When evaluated on translated source data, TS-K_S shows the same trend of improving with the number of samples, but its performance with 5 relational samples is slightly worse than MS-D_S.

Conclusion

Knowledge translation is an important task towards knowledge reuse where the knowledge in the source schema

[2] http://alchemy.cs.washington.edu/data/uw-cse/.

[3] http://alchemy.cs.washington.edu/alchemy1.html

needs to be translated to a semantically heterogeneous target schema. Different from data integration and transfer learning, knowledge translation focuses on the scenario that the data may not be available in both the source and target. We propose a novel probabilistic approach for knowledge translation by combining probabilistic graphical models with schema mappings. We have implemented an experimental knowledge translation system and evaluated it on two real datasets for different prediction tasks. The results and comparison with baselines show that our approach can obtain comparable accuracy without data.

The proposed log-linear models, such as Markov random fields and Markov logic networks, already cover most of common types of knowledge used in data mining. In the future work, we will extend our approach to the knowledge types which are harder to represent as log-linear models, such as SVMs and nearest neighbor classifiers. It might require a specialized probabilistic representation.

Acknowledgement This research is funded by NSF grant IIS-1118050.

References

Caragea, D.; Zhang, J.; Bao, J.; Pathak, J.; and Honavar, V. 2005. Algorithms and software for collaborative discovery from autonomous, semantically heterogeneous, distributed information sources. In *Proceedings of the 16th International Conference on Algorithmic Learning Theory*, ALT'05, 13–44.

Davis, J., and Domingos, P. 2009. Deep transfer via second-order Markov logic. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 217–224.

Domingos, P., and Lowd, D. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool.

Dong, X.; Halevy, A. Y.; and Yu, C. 2007. Data integration with uncertainty. In *VLDB '07: Proceedings of the 33rd International Conference on Very Large Data Bases*, 687–698. VLDB Endowment.

Dong, X. L.; Halevy, A.; and Yu, C. 2009. Data integration with uncertainty. *The VLDB Journal* 18(2):469–500.

Dou, D.; Qin, H.; and Liu, H. 2011. Semantic translation for rule-based knowledge in data mining. In *Proceedings of the 22nd International Conference on Database and Expert Systems Applications*, volume Part II of *DEXA'11*, 74–89.

Fang, M.; Yin, J.; Zhu, X.; and Zhang, C. 2015. TrGraph: Cross-network transfer learning via common signature subgraphs. *IEEE Trans. Knowl. Data Eng.* 27(9):2536–2549.

Getoor, L., and Taskar, B., eds. 2007. *Introduction to Statistical Relational Learning*. Adaptive Computation and Machine Learning. The MIT Press.

Kindermann, R.; Snell, J. L.; et al. 1980. *Markov random fields and their applications*, volume 1. American Mathematical Society Providence, RI.

Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Lenzerini, M. 2002. Data integration: A theoretical perspective. In *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '02, 233–246.

Lowd, D., and Davis, J. 2014. Improving markov network structure learning using decision trees. *Journal of Machine Learning Research* 15(1):501–532.

Lowd, D., and Domingos, P. 2007. Efficient weight learning for Markov logic networks. In *In Proceedings of the Eleventh European Conference on Principles and Practice of Knowledge Discovery in Databases*, 200–211.

Lowd, D., and Rooshenas, A. 2015. The libra toolkit for probabilistic models. *arXiv preprint arXiv:1504.00110*.

Mihalkova, L.; Huynh, T.; and Mooney, R. J. 2007. Mapping and revising Markov logic networks for transfer learning. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, volume 1 of *AAAI'07*, 608–614.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10):1345–1359.

Park, B.-H., and Kargupta, H. 2002. Distributed data mining: Algorithms, systems, and applications. In Ye, N., ed., *The Handbook of Data Mining*. Lawrence Erlbaum Associates. 341–358.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Poole, D. 2003. First-order probabilistic inference. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, 985–991. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Poon, H., and Domingos, P. 2006. Sound and efficient inference with probabilistic and deterministic dependencies. In *Proceedings of the 21st National Conference on Artificial Intelligence*, volume 1 of *AAAI'06*, 458–463.

Rahm, E., and Bernstein, P. A. 2001. A survey of approaches to automatic schema matching. *The VLDB Journal* 10(4):334–350.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine Learning* 62:107–136.

Russell, S. J., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition.

Yang, Q.; Chen, Y.; Xue, G.-R.; Dai, W.; and Yu, Y. 2009. Heterogeneous transfer learning for image clustering via the social web. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, volume 1 of *ACL '09*, 1–9.

Ye, J.; Cheng, H.; Zhu, Z.; and Chen, M. 2013. Predicting positive and negative links in signed social networks by transfer learning. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, 1477–1488.