ECAI 2010
H. Coelho et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-606-5-937

Adaptive Markov Logic Networks: Learning Statistical Relational Models with Dynamic Parameters

Dominik Jain and **Andreas Barthels** and **Michael Beetz**[1]

Abstract. Statistical relational models, such as Markov logic networks, seek to compactly describe properties of relational domains by representing general principles about objects belonging to particular classes. Models are intended to be independent of the set of objects to which these principles can be applied, and it is assumed that the principles will soundly generalize across arbitrary sets of objects. In this paper, we point out limitations of models that seek to represent the corresponding principles with a fixed set of parameters and discuss the conditions under which the soundness of fixed parameters is indeed questionable. We propose a novel representation formalism called *adaptive Markov logic networks* to allow more flexible representations of relational domains, which involve parameters that are dynamically adjusted to fit the properties of an instantiation by phrasing the model's parameters as functions over attributes of the instantiation at hand. We empirically demonstrate the value of our learning and representation system on a simple but well-motivated example domain.

1 INTRODUCTION

The field of statistical relational learning (SRL), which has been gaining more and more attention in recent years, deals with the learning of generalized probability distributions over object attributes and relations between objects. The key assumption is that one can extract general principles about objects having similar properties (e.g. about objects belonging to the same class) from data, represent these principles declaratively in a statistical relational model and consequently apply them to new domains of discourse (i.e. collections of objects belonging to the classes under consideration) in order to obtain a concrete model of a probability distribution that should be sound with respect to the dependencies being represented – both qualitatively and quantitatively. This is often referred to as *shallow transfer*.

Representation formalisms underlie the assumption that the parameters that were once learned with a (sufficiently large) training database hold for every other domain, regardless of its size (or other properties). Unfortunately, this assumption does not always hold. One particularly common trait of relational domains that typically results in a violation of the assumption is the presence of cardinality restrictions, i.e. restrictions that constrain the number of relational partners of an entity. Such restrictions are inherent properties of relational domains (as is evident from their abundance in e.g. entity-relationship models [1]) and therefore need to be integrated into any statistical model that seeks to capture relational properties accurately. While, in principle, the cardinality constraints themselves can be integrated into some of the more expressive statistical relational models (including Markov logic networks), their complex interactions with probabilities pertaining to the constrained relations are largely neglected, severely limiting the models' capability of performing *link prediction* – one of the key tasks in SRL – within reasonable bounds of accuracy as domain size varies.

We demonstrate these problems using a very simple scenario, in which we consider students taking courses at a university. We differentiate students enrolled in the Bachelor's program and the Master's program, as well as beginner-level courses and advanced-level courses. Assuming that students of any program can pick courses of either level while, however, being inclined to pick courses that are appropriate with respect to their level of advancement, the probability of a student taking a course will depend on the type of the student and the course. Assuming that every student is required to take a particular number of courses, one can typically observe that if a statistical relational model is trained on a single database with a particular number of courses and students, the probabilities indicated for the relation linking students to the courses they take are specific to the cardinalities that were present in the training data. Therefore, link prediction tends to yield unsatisfactory results when the learned model is applied to domains that differ (in terms of size) from the training database.

In this work, we seek to solve the corresponding problems. Our contributions are along two dimensions: First, we provide a discussion of the way in which generalization in statistical relational models is to be viewed, creating an awareness for possible limitations of current approaches based on static parameters. Second, we introduce a novel representation formalism that addresses the issues that we identified, which we base upon one of the most expressive formalism to have been proposed thus far, Markov logic networks (MLNs). As secondary contributions, we address general issues pertaining to the learning of MLN parameters that will generalize across domains of variable size and the representation and handling of cardinality restrictions.

The next section begins by motivating the need for adaptive models. In Section 3, we briefly review the fundamentals of MLNs, which we proceed to extend to adaptive MLNs in Section 4 in order to remove the limitations imposed by fixed-parameter models. In Section 5, we provide an empirical evaluation. We mention related work in Section 6 before concluding with Section 7.

2 LIMITS OF SHALLOW TRANSFER

For a model to accurately represent any particular domain of discourse (instance of a scenario), it is a prime requirement that all the principles which are being represented in the model are in fact *independent* of properties of the domain – in particular, its size. As soon as this is not truly the case, any model is bound to represent probability distributions that may diverge arbitrarily from the true distribution

[1] Technische Universität München, Germany, email: jain@cs.tum.edu

as we apply it to domains that differ from the domain it was trained with (in terms of size or otherwise).

A statistical relational (meta-)model is applied by instantiating it for a particular *domain of discourse*, i.e. in the case of a typed logic, a set of (non-empty) sets of constants referring to entities in the world, one set for each class. The instantiation typically involves a mere reproduction of substructures that appear in the meta-model, i.e., more precisely, the repeated application of templates in order to obtain an actual *ground model* of a probability distribution that is specific to the domain of discourse. This probability distribution is typically represented as a probabilistic graphical model, e.g. a Bayesian network or, as in the case of MLNs, a Markov random field.

We can view any statistical relational model as a representation of a stochastic process that embodies some well-defined characteristics about a particular scenario. Whether or not the application of rigid templates containing a fixed set of parameters can be considered sound regardless of the nature of the stochastic process that is to be represented is highly dependent on the way in which we interpret the domains to which we apply the model. Since one generally seeks to minimize statistical errors that result from small sample sizes, models are typically trained using large training databases. If we now instantiate a trained model for a smaller set of objects, the probability distribution indicated by the ground model can certainly be considered sound (even if it contains parameters that are in fact specific to the training data), if we view that set of objects as a subset of the set of objects in the training database, i.e. if we implicitly assume that the instantiation contains additional objects so as to reach the number of objects present in the training database. In this mode of application, the training database must be viewed as the single domain of discourse in its entirety: Any smaller instantiation is merely an excerpt of it, and larger instantiations should be considered as invalid. In some cases, this may indeed be sensible, yet clearly, it ultimately defeats the very purpose of statistical relational models to describe general principles that should be applicable to arbitrary instantiations, since what is really being captured is but a (propositional) model that describes the training database, and any relational aspects are reduced to mere syntactic sugar.

If, however, the training database is not to be viewed as the sole universe of objects, then the nature of the stochastic process that we seek to model may dictate that parameters must indeed change with domain size. A straightforward example of such a case is a process that contains relations that are subject to cardinality restrictions and where any world that is generated by it should be considered as *self-contained*. In other words, consider a process that demands that the number of objects to which particular objects are to be related is fixed or otherwise constrained, and, for each object, all the relational partners are always part of the universe of objects being generated. As we will see, this rather simple condition is sufficient for a model with fixed parameters to fail to generalize as expected.

The presence of cardinality restrictions in a self-contained domain implies that if the set of potential relational partners changes, then so must the probability of there being a relation. For instance, if we consider the relation *isParentOf*(x, y), which holds if x is a parent of y, then, provided that every child's parents are in fact part of the instantiation, the probability of there being a relation between an arbitrary child-parent pair decreases as the number of parents in the domain increases (it is $2/n$ if n is the number of potential parents). Of course, without further restrictions, an explicit representation of the cardinality constraint that limits the number of parents to 2 is sufficient in order to correct the marginal probability of the *isParentOf* relation. Since Markov logic subsumes first-order logic, cardinality restrictions can easily be formulated: For an arbitrary binary relation *rel*(x, y), we could state that for each x there should be exactly c objects to which x is to be related as follows:

$$\exists\, y_1, \ldots, y_c.\ \bigwedge_{i=1}^{c} \left(\mathit{rel}(x, y_i) \wedge \bigwedge_{j=1}^{i-1} y_i \neq y_j \right) \wedge \\ \neg\exists\, y_{c+1}.\ \mathit{rel}(x, y_{c+1}) \wedge \bigwedge_{j=1}^{c} y_{c+1} \neq y_j \qquad (1)$$

While such constraints succeed at correcting marginal probabilities of constrained relations, they do not suffice in cases where the relation is in fact dependent on, for example, attributes of the related objects (a quite common case, as we usually imagine stochastic processes to first generate objects and then establish relations between them based on their properties). The reason is that the introduction of such dependencies requires that, for each configuration of the attributes the relation depends on, we are required to capture the proper ratio between the case where the relation occurs and where it does not occur. Modelling only the ratios between cases where the relation is to hold, which are indeed size-invariant, is, unfortunately, not sufficient.

To illustrate this, consider the simplest of cases, where we have two types of entities, children and parents, that (with probability $1/2$) have some boolean property which influences the likelihood of there being a relation. We know that each child must be related to exactly two parents. We could reasonably represent this scenario using the marginal distribution of the property for parents and children, the conditional distribution of the relation given the properties of both entities, plus the cardinality restriction. Assume that the probability p_1 with which a child is related to a parent given that both have the property or both do not is four times as high as the probability p_2 where one has the property and the other does not. Given a stochastic process that generates data reflecting the above, we can train models on databases of variable size and then instantiate the obtained models for a particular set of objects we are interested in. Exemplarily, assume that there is a particular child C and a parent P who both have the property as well as another child and three further parents about whom we do not know anything. We ask for the probability q of C being related to P in comparison to C being related to one of the three other parents (q'). Table 1 summarizes results we obtained for such an experiment. Note that even though we explicitly represented the cardinality constraint (which implies $q + 3q' = 2$) and the ratio $p_1 : p_2$ is indeed $4 : 1$ in each case, the probability q varies depending on the training database and, most importantly, diverges considerably from the true result – simply because the ratios $p_1 : (1 - p_1)$ and $p_2 : (1 - p_2)$ can be regarded as arbitrary with respect to the instantiation that we considered.

Table 1: Parameters obtained from training databases of various sizes and corresponding inference results, where M_N indicates a model derived from a database with N children and $2N$ parents.

	M_{10}	M_{100}	M_{200}	*Correct/Extrapolation*
p_1	0.1600	0.0160	0.0080	0.8000
p_2	0.0400	0.0040	0.0020	0.2000
q	0.6237	0.6044	0.6034	0.8290
q'	0.4588	0.4652	0.4655	0.3903

3 MARKOV LOGIC NETWORKS

Markov logic networks combine first-order logic with uncertainty [9]. An MLN L is given by a set of pairs $\langle F_i, w_i \rangle$, where F_i is a formula in first-order logic and w_i is a real number, the weight of formula F_i. For each finite domain of discourse D (set of constants), an MLN L defines a *ground Markov random field* $M_{L,D}$ as follows:

1. $M_{L,D}$'s set of variables X contains one boolean variable for each possible grounding of each predicate appearing in L.
2. $M_{L,D}$ contains one feature for each possible grounding of each formula F_i in L. The value of this feature is 1 if the ground formula is true, and 0 otherwise. The weight of the feature is w_i.

$M_{L,D}$ specifies a probability distribution over the set of possible worlds, i.e. the set of possible assignments of truth values to each of the ground atoms in X, as follows,

$$P_w(X = x) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(x)\right) \tag{2}$$

where $n_i(x)$ denotes the number of true groundings of F_i in x and Z is a normalization constant.

The parameters of an MLN (the weights w_i) are usually learned using MAP estimation or maximum likelihood. In the latter case, one simply maximizes the (pseudo-)likelihood of a training database x (i.e. a possible world that defines, for some domain D, the truth values of all the ground atoms in the set X specific to D).

4 ADAPTIVE MARKOV LOGIC NETWORKS

We now introduce a generalization of Markov logic networks that addresses the issues described in Section 2. Since we want model parameters to be dependent on attributes of an instantiation, we need to formalize these attributes: Let $\{A_1, \dots, A_k\}$ be the set of attributes that we consider. $\mathcal{A} := \mathrm{dom}(A_1) \times \cdots \times \mathrm{dom}(A_k)$ is thus the set of possible attribute configurations. We define an *adaptive Markov logic network* (AMLN) as a set $\mathcal{L}$ of pairs $\langle F_i, W_i \rangle$, where F_i is a formula in first-order logic and $W_i : \mathcal{A} \to \mathbb{R}$ is a function that maps from the vector of attributes of the instantiation to the parameter domain. An AMLN can therefore be regarded as a template for the construction of an MLN: For any domain D with attribute vector $a_D \in \mathcal{A}$, we obtain a Markov logic network $L_{\mathcal{L},D} = \{\langle F_i, W_i(a_D)\rangle\}$ specific to D, which we can then apply in order to perform inference.

In the remainder of this work, the attributes A_i we consider are strictly cardinalities of sets of objects as they appear in the instantiation, i.e. $\mathrm{dom}(A_i) = \mathbb{N}$, but it is conceivable that parameters could be dependent on other attributes. For instance, parameters could depend on the point in time for which we instantiate the model: We could learn from several historical records of, say, a social network, learn how dependencies changed over time and then instantiate the adaptive model for a point in time for which we do not (yet) have any data.

4.1 PARAMETER LEARNING

The key to the learning of AMLNs that will indeed generalize across domains with varying attribute configurations is to adjust the learning algorithms to learn the weight functions that we introduced above rather than scalar values. This naturally requires us to learn from multiple training databases; the combination of several training databases into a single, very large training database of i.i.d. data is clearly not adequate. We need to explicitly consider the effects of changing the domain of discourse.

4.1.1 *Constrained Learning from Individual Databases*

First of all, however, we need to take into consideration the fact that parameter learning in MLNs is, in general, an ill-posed problem [5]. Consider the MLN structure shown in Table 2. (We use a notation in which functional relations are declared by suffixing functionally determined arguments of a relation with an exclamation mark.) The predicates *sT* and *cT* stand for *student type* and *course type* respectively. The MLN that is shown is thus a basic model of our example scenario where the relation *takes* depends on the attributes of the related objects. The ill-posedness of parameter learning stems from the underdeterminism that allows us to obtain, for particular cardinalities of the sets of students and courses, precisely the same probability distribution using an infinite number of (non-equivalent) weight vectors, as the choice of δ in Table 2 is arbitrary. For any real δ, the MLNs with weight vector w and w' are equivalent. The effect of modifying the weight of the unit clause $sT(s, BSc)$ can be cancelled out by applying the inverse modification to a set M of mutually exclusive and exhaustive formulas within which that unit clause appears – scaled, however, using the ratio between the number of groundings of the unit clause and the number of groundings of each of the formulas in M. With N_c and N_s as the number of courses and the number of students in the domain respectively, the ratio is $N_s/(N_s \cdot N_c) = 1/N_c$. It should be clear, however, that, for $\delta \neq 0$, if N_c changes as we move to another domain of discourse, so does the implied probability distribution.

If unit clauses such as $sT(s, BSc)$ are, as suggested in [9], indeed to be used to capture marginal probabilities, we must therefore ensure that, for a group of mutually exclusive and exhaustive unit clauses (such as formulas 1 and 2 in Table 2), the ratio between any two exponentiated formula weights within the group matches the empirical relative frequencies in the training database. For instance, if the fraction of Bachelor students is 2/3, we should require that $\exp(w_1)/\exp(w_2) = (2/3)/(1/3) = 2$.

Table 2: MLN structure exhibiting underdeterminism

predicate declarations	*domain definitions*
sT(*student*, *studentType!*) *cT*(*course*, *courseType!*) *takes*(*student*, *course*)	$studentType = \{BSc, MSc\}$ $courseType = \{Beg, Adv\}$

formula	w	w'
$sT(s, BSc)$	w_1	$w_1 + \delta$
$sT(s, MSc)$	w_2	w_2
$cT(c, Beg)$	w_3	w_3
$sT(c, Adv)$	w_4	w_4
$takes(s, c) \wedge sT(s, BSc) \wedge cT(c, Adv)$	w_5	$w_5 - \delta/N_c$
$\neg takes(s, c) \wedge sT(s, BSc) \wedge cT(c, Adv)$	w_6	$w_6 - \delta/N_c$
$takes(s, c) \wedge sT(s, BSc) \wedge cT(c, Beg)$	w_7	$w_7 - \delta/N_c$
$\neg takes(s, c) \wedge sT(s, BSc) \wedge cT(c, Beg)$	w_8	$w_8 - \delta/N_c$
$takes(s, c) \wedge sT(s, MSc) \wedge cT(c, Adv)$	w_9	w_9
$\neg takes(s, c) \wedge sT(s, MSc) \wedge cT(c, Adv)$	w_{10}	w_{10}
$takes(s, c) \wedge sT(s, MSc) \wedge cT(c, Beg)$	w_{11}	w_{11}
$\neg takes(s, c) \wedge sT(s, MSc) \wedge cT(c, Beg)$	w_{12}	w_{12}

Therefore, in order to eliminate the underdeterminism, we propose a controllable preprocessing stage during parameter learning. A straightforward choice of weights that guarantees the required ratios is to set the weight of a unit clause to the logarithm of its relative frequency in the data. Of course, only the weights of unit clauses pertaining to a priori independent variables should be fixed in this way. In our example, we assume that *takes* is not subject to a fixed marginal distribution and therefore we would not – even if a corresponding unit clause had been included in our model – have wanted its weight to be determined by the preprocessing stage. We therefore use explicit declarations in the model structure to determine the set of formulas that the preprocessing stage is to be applied to.

Following the preprocessing stage, which reduces the dimension of the learning problem to the respective projection of the weight vector, standard parameter learning is performed in order to learn the remaining weights. We use BFGS to optimize the pseudo-likelihood of the possible world embodied by the training database.

4.1.2 Learning from Multiple Databases

We now address the issue of learning dependencies of parameters on attributes of the instantiation from multiple databases, using the learning procedure outlined above as a subroutine. Let $\{D_1, \ldots, D_k\}$ be the set of training databases. The first step is to apply, to each training database D_j, the learning procedure described in the previous section. This yields a vector of weights $\hat{w}_i$ for each formula appearing in the AMLN structure,

$$\hat{w}_i = \left((\hat{w}_i)_1 \quad (\hat{w}_i)_2 \quad \cdots \quad (\hat{w}_i)_k \right) \tag{3}$$

Since we consider models where the objects are typed, each training database contains a set of objects for each of the types (object classes). Let $(D_j)_l$ be the set of objects belonging to the l-th type in the j-th training database. We call $(D_j)_l$ a *subdomain* of D_j.

Reducing the set of domain attributes to cardinalities of subdomains, we consider the weight function W_i of the i-th formula to be a function of sizes of subdomains referenced in F_i, i.e. its signature is given by $W_i : \mathbb{N}^{s_i} \to \mathbb{R}$, where s_i is the number of subdomains referenced in F_i. We express each weight function as a linear combination of a set of basis functions. First, we give a brief motivation on how to reasonably choose these basis functions. Observe that Equation (2), which describes the probability of a possible world given a ground Markov random field, can be rewritten as

$$P_w(X = x) = \tfrac{1}{Z} \exp\left(\textstyle\sum_i w_i n_i(x)\right) = \tfrac{1}{Z} \textstyle\prod_i \omega_i^{n_i(x)} \tag{4}$$

with $\omega_i = \exp(w_i)$. To capture the dependency on subdomain sizes, we view ω_i as a function over some size m (i.e. $m = |(D_j)_l|$ for some j, l). If we want the above product to be capable of representing a factorization that includes relative frequencies α/m (e.g. 6 out of 24), our basis functions need to be capable of describing $\omega_i(m) := \alpha m^{-1}$. By choosing as basis functions

$$\phi_0 \equiv 1, \qquad \phi_1(m) := \log(m), \qquad \phi_2(m) := m \tag{5}$$

we are capable of learning any monomial or exponential ω_i:

- $W_i(m) = \log(\omega_i(m)) = \log(\alpha m^c) = \log\alpha + c\log m$ (6)
- $W_i(m) = \log(\omega_i(m)) = \log(\alpha c^m) = \log\alpha + m\log c$ (7)

Similarly, should a formula depend on more than one subdomain, frequencies over products of subdomain sizes can be handled as follows:

$$\begin{aligned} W_i(m_1, m_2) &= \log(\omega_i(m_1, m_2)) = \log(\alpha m_1^{c_1} m_2^{c_2}) \\ &= \log\alpha + c_1 \log m_1 + c_2 \log m_2 \end{aligned} \tag{8}$$

Learning faster-growing classes of functions is unlikely to be advantageous. One might add more slow-growing basis functions, but the gain is questionable, especially since adding too many basis functions is likely to result in overfitting.

To learn the actual weight function W_i, we need to find coefficients $\alpha^{(i)}_{\cdot,\cdot}$ such that the combined basis functions approximate the learned weights $(\hat{w}_i)_j$ as well as possible. For each formula F_i and training database D_j, we have

$$(\hat{w}_i)_j = \phi_0 \alpha^{(i)}_{0,\phi_0} + \sum_{l \in \mathcal{I}_i} \sum_{k=1}^{2} \phi_k(|(D_j)_l|) \alpha^{(i)}_{l,\phi_k} + (e_i)_j \tag{9}$$

where $\mathcal{I}_i$ is the set of indices of domains appearing in formula F_i and $(e_i)_j$ is an error term. The overall goal is to minimize the quadratic norm of the vector of errors $E_i = \left((e_i)_j\right)$. Equation (9) is a system of linear equations. Let x_i be the vector of coefficients, i.e. $x_i = (\alpha^{(i)}_{\cdot,\cdot})$, and let A_i be the matrix of basis functions evaluated at the relevant subdomain sizes, sequentially taken from all training databases. Assuming canonical ordering of entries in the matrix and vector, we can rewrite equation (9) as $A_i x_i \approx \hat{w}_i$ and use standard linear curve fitting methods to obtain an optimal approximation vector x_i that parameterizes our weight function W_i.

4.2 EXPLICIT CARDINALITY CONSTRAINTS

In the examples that we consider further on, relations are subject to cardinality restrictions. Since the use of existential quantification to express cardinality restrictions as in Equation (1) is usually prohibitively expensive in practice – owing to the conjunctive normal form (CNF) conversion, which, unfortunately, results in an exponential blow-up of the representation[2] – we extended our implementation of AMLNs with a language construct that represents such restrictions explicitly,

$$\textit{count}(\textit{rel}(x_1, \ldots, x_n) \,|\, x_{i_1}, \ldots, x_{i_m}) \in S \tag{10}$$

where $m \leq n$ and $S \subset \mathbb{N}$, the semantics being that if the parameters that are given by the index set $\{i_1, \ldots, i_m\}$ are bound to some fixed vector of constants, the number of bindings for the remaining parameters for which the relation holds true is required to be in S.

In the context of MLNs, a (weighted) *count* constraint defines, for each possible grounding of the fixed parameters, a feature in a ground Markov random field that corresponds to the clique connecting the variables (ground atoms) that one obtains by grounding the remaining parameters of the relation predicate; this is analogous to the clique implied by Equation (1).

In our experiments, we use the inference algorithm MC-SAT [8], as it proved to be most robust. Fortunately, the algorithm is easily extended to support our notion of explicit cardinality constraints. MC-SAT is a slice sampler that uses one auxiliary variable per constraint. The initial state (assignment of truth values to each of the ground atoms) must be one that satisfies all hard constraints. Then, in each step, the auxiliary variables are resampled given the current state. Every setting of the auxiliary variables implies a subset $\mathcal{M}$ of the ground model's constraints that must be satisfied and, in particular, a uniform distribution over the subset of possible worlds in which these constraints are satisfied. Therefore, the next state is chosen by uniformly sampling from the subset of possible worlds where the constraints in $\mathcal{M}$ are satisfied. For this purpose, MC-SAT uses the SampleSAT algorithm [10], which extends the WalkSAT algorithm by injecting randomness via simulated annealing-type moves, allowing near-uniform samples to be obtained.

With respect to MC-SAT, a *count* constraint can be treated just like any regular logical constraint; it has a regular weight and we can easily check whether a given state satisfies it. Since MC-SAT requires weights to be positive, we may need to negate *count* constraints beforehand. (Note that in an MLN, a formula F with weight w has precisely the same as effect as $\neg F$ with weight $-w$.) To negate a *count* constraint, we simply need to invert the set of counts S, i.e. we use as the new set of counts the complement $\mathbb{N} \setminus S$.

[2] Although a CNF conversion is not an unavoidable precondition for an implementation of Markov logic networks, it is generally necessary to support inference methods such as MC-SAT.

Furthermore, we need to make changes to the SampleSAT algorithm in order to enable it to find assignments that satisfy *count* constraints. Simulated annealing moves are largely unaffected by the introduction of *count* constraints; we only need to incorporate into the delta cost of the random move (which is used to probabilistically decide whether the move is actually taken) the number of *count* constraints that would become unsatisfied or satisfied as a result of the move. For WalkSAT moves, we pick one of the yet unsatisfied constraints and then flip the truth values of one or more ground atoms that will satisfy the constraint and cause as few other constraints in $\mathcal{M}$ to be unsatisfied as possible. For an unsatisfied *count* constraint, the current state will either satisfy too few atoms from the set Y of ground atoms appearing in the ground *count* constraint or too many, i.e. we must set one or more ground atoms that are currently false to true or vice versa. In either case, we greedily choose the ground atom(s) with the most favourable delta cost of newly satisfied minus newly unsatisfied constraints and flip their truth values. If the current number of true ground atoms from Y is between two values from the constraint's set of counts S, we consider the cost of both variants and choose the direction in which to move accordingly.

5 EVALUATION

In the following, we demonstrate the problems pertaining to shallow transfer that we described above, illustrating how our methodology solves them. To understand the gravity of these problems, it is sufficient to look at the very simple example scenario that we previously introduced: Consider the stochastic process characterized by Table 3 for this scenario. By clearly defining the properties of the stochastic process, we can easily evaluate any model's ability to reflect these properties – once it has been appropriately trained on training databases generated by the process.

Table 3: Properties of the stochastic process

cardinalities:
count($cT(c, Beg)$) $= 6$
count($cT(c, Adv)$) $= \|\text{course}\| - 6$
count($takes(s, c) \mid s$) $= 6$

formula	*p*
$sT(s, BSc)$	$2/3$
$sT(s, MSc)$	$1/3$

	course participation preferences of a	
	BSc	*MSc*
Beg	$4/5$	$1/3$
Adv	$1/5$	$2/3$

The process states that the program that any given student is enrolled in is Bernoulli-distributed; with probability $2/3$, a student is enrolled in the BSc program. Furthermore, every student is required to take exactly six courses out of the ones on offer: There are always six beginner-level courses; all other courses are advanced-level courses. The choice of courses taken by a student depends on both the student's status and the courses' levels. For example, if a course is taken by an MSc, it is four times as likely to be an advanced course than a beginner-level course. Since all students take exactly six courses, the probability of a course being taken by some student varies with domain size. It is important to note that the right part of Table 3 is *not* a conditional distribution that could simply be included in a directed model, because the probability of a course having a particular type is affected by all the students' relations to the course. In the following, the goal is to learn and represent the properties of this stochastic process in such a way that the effects resulting from an application of the model to instantiations of variable size are taken into account.

As our MLN model structure, we used the structure shown in Table 2, extended only with the following hard cardinality constraints, which correspond directly to the properties of the process:

$$\begin{aligned} &\textit{count}(\textit{takes}(s, c) \mid s) \in \{6\}. \\ &\textit{count}(\textit{cT}(c, \textit{Beg})) \in \{6\}. \end{aligned} \tag{11}$$

Note that the structure in Table 2 on its own would be ideal for a process where there is no cardinality restriction on the relation *takes*, for it is capable of representing precisely a direct factorization of the full-joint probability distribution as it would appear in a model derived via knowledge-based model construction (KBMC).

We generated a series of training databases with a varying number of courses (it ranged from 12 to 36), while the number of students was fixed at 45.

5.1 PARAMETER LEARNING

To evaluate the learning of weight functions, we trained our AMLN on a set of databases of five different sizes, using multiple databases for each size to compensate for the stochasticity of our process model. In total, our training databases contained data on 3,375 students and 1,875 courses.

The appropriateness of the basis functions that we proposed in Section 4.1.2 is confirmed by the exemplary results shown in Figure 1a, which indicates that even for domain sizes we did not explicitly train for, the weights are near-optimal. Thus, the dynamics of the stochastic process appear to be adequately represented in the weight function that was learned.

5.2 LINK PREDICTION

Having learned an AMLN for our example domain, the next step is to evaluate it in terms of how well the model reflects the properties of the process for any given domain size. We compare our results to standard MLNs trained on a single database exhibiting *precisely* the expected values of our stochastic process (we used a deterministic generator process to guarantee this), and its size was equivalent to that of the unification of all training databases that were used to induce the AMLN. All models used the same structure including cardinality constraints.

Since the task of link prediction is most critical within the context of self-contained worlds and constrained relations, we inferred the probability of a student taking a particular course given the attributes of both the course and the student – for domains containing a variable number of additional courses (about which we do not provide further evidence). The results are summarized in Figure 1b. The AMLN accurately represents the process property which demands that as the number of courses increases, the probability of an *MSc* student taking any given advanced-level course decreases appropriately (it is $\frac{2}{3} \cdot 6/(N_c - 6)$ if N_c is the number of courses in the domain). The standard MLN, on the other hand, does not capture this fundamental process property. Given what we have learned thus far, this was to be expected (since it was trained on a database containing a number of courses that is far greater than the number for which the model was instantiated), yet the degree to which the predictions diverge from the desired results is notable (the relative error exceeds 90%).

It is interesting to note that our AMLN learning approach is capable of capturing properties of the underlying stochastic process even if the model structure is incomplete. In particular, if we remove all cardinality constraints from the model, the AMLN still correctly estimates at least the expected values that these cardinality constraints imply: Applying link prediction to evidence databases containing ten students and a varying number of courses, providing information on all attribute values as they were generated by the stochastic process

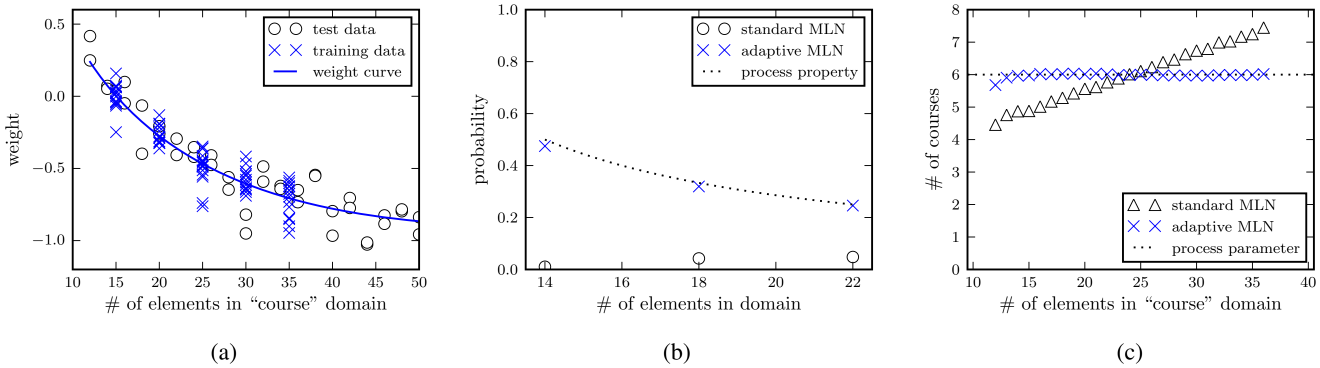

Figure 1: (a) weight curve of $takes(s,c) \wedge sT(s, MSc) \wedge cT(c, Adv)$; (b) probability of a particular Master student taking a particular advanced course given only the existence of further courses; (c) expected number of courses taken by a student depending on the number courses offered

model, we observe that the expected number of courses taken by a student is predicted at almost exactly six by the AMLN for all domain sizes, as shown in Figure 1c. A standard MLN, which was induced using a (perfect) training database containing 24 students, predicts the desired outcome only for the domain size it was trained for.

5.3 CLASSIFICATION

Apart from probabilities pertaining to links, we can (as long as the *entire* relational skeleton is given) expect standard MLNs to accurately capture dependencies between class attributes and any predictor attributes. Therefore, the classification performance of MLNs and AMLNs (with known relations) is equivalent. As soon as, however, even a few links are left unknown, classification results are greatly affected. We empirically confirmed this in our experiments, but for reasons of brevity, we do not report the results.

6 RELATED WORK

Cardinality restrictions have found their way into most areas of computer science that deal with (natural) relations in one way or another, be it the modelling of entity relationships for database design [1] or the logical modelling of real-world concepts and relations in description logics [2]. Soft constraints on cardinalities, i.e. distributions over counts, can, for instance, explicitly be expressed in the probabilistic description logic PClassic [6] and were also considered in [7]. Cardinality-based features of graphical models were previously defined in [4] in order to reduce the complexity of inference, albeit in an entirely different context. Advances in non-parametric models (e.g. [3]), though concerned with issues pertaining to cardinalities, also have a different focus. The impact of cardinality restrictions on the generalization of statistical relational models across domains of variable size has, to the best of our knowledge, so far not been addressed. Dynamic parameter adjustments in probabilistic logical models were previously proposed in [5] in order to model size-invariant domain properties – in particular, marginal probabilities that must not change with domain size. With AMLNs, we address the inverse problem of modelling precisely how probabilities must change with domain size (or other attributes of an instantiation).

7 CONCLUSION

In this paper, we introduced statistical relational models with dynamic parameters: adaptive Markov logic networks (AMLNs).[3] The dynamic adjustment of parameters is, as indicated by our experiments, an important concept in order to enable models to generalize across domains with varying attributes (in particular, attributes pertaining to the number of objects under consideration). As we have shown, statistical models with a fixed vector of parameters do not in general succeed in capturing the general principles that in fact apply to even conceptually simple relational domains, producing arbitrarily large errors in link prediction tasks. In contrast, the representation that we proposed and the learning mechanism that we devised are capable of handling the stochastic effects that result from varying attributes of the domain of discourse and thus enable a far greater class of stochastic processes to be accurately represented.

While our current approach is based on Markov logic networks, our methodology can be straightforwardly adapted to other formalisms that have been devised in the field of statistical relational learning.

Acknowledgements. This work was partly supported by the CoTeSys (Cognition for Technical Systems) cluster of excellence.

REFERENCES

[1] J.-R. Abrial, 'Data Semantics', in *IFIP TC2 Working Conference on Data Base Management*, eds., J. W. Klimbie and K. L. Koffeman, pp. 1–59. Elsevier Science Publishers (North-Holland), (April 1974).

[2] Franz Baader and Ulrike Sattler, 'Expressive Number Restrictions in Description Logics', *J. Log. Comput.*, **9**(3), 319–350, (1999).

[3] Peter Carbonetto, Jacek Kisynski, O De Freitas, and David Poole, 'Nonparametric Bayesian Logic', in *UAI*. UAI, (2005).

[4] Rahul Gupta, Ajit A. Diwan, and Sunita Sarawagi, 'Efficient Inference with Cardinality-Based Clique Potentials', in *ICML*, pp. 329–336, (2007).

[5] Dominik Jain, Bernhard Kirchlechner, and Michael Beetz, 'Extending Markov Logic to Model Probability Distributions in Relational Domains', in *KI-2007*, pp. 129–143, (2007).

[6] Daphne Koller, Alon Levy, and Avi Pfeffer, 'P-classic: A tractable probabilistic description logic', in *Proceedings of AAAI-97*, pp. 390–397, (1997).

[7] Avi Pfeffer, Daphne Koller, Brian Milch, and Ken T. Takusagawa, 'Spook: A system for probabilistic object-oriented knowledge representation', in *UAI*, pp. 541–550. Morgan Kaufmann, (1999).

[8] Hoifung Poon and Pedro Domingos, 'Sound and Efficient Inference with Probabilistic and Deterministic Dependencies', in *AAAI*. AAAI Press, (2006).

[9] Matthew Richardson and Pedro Domingos, 'Markov Logic Networks', *Mach. Learn.*, **62**(1-2), 107–136, (2006).

[10] Wei Wei, Jordan Erenrich, and Bart Selman, 'Towards Efficient Sampling: Exploiting Random Walk Strategies', in *AAAI*, pp. 670–676, (2004).

[3] Source code is freely available from http://ias.cs.tum.edu/research/probcog