Mach Learn (2008) 73: 3–23
DOI 10.1007/s10994-008-5079-1

Structured machine learning: the next ten years

Thomas G. Dietterich · Pedro Domingos · Lise Getoor · Stephen Muggleton · Prasad Tadepalli

Received: 25 October 2007 / Accepted: 2 February 2008 / Published online: 20 August 2008

Abstract The field of inductive logic programming (ILP) has made steady progress, since the first ILP workshop in 1991, based on a balance of developments in theory, implementations and applications. More recently there has been an increased emphasis on Probabilistic ILP and the related fields of Statistical Relational Learning (SRL) and Structured Prediction. The goal of the current paper is to consider these emerging trends and chart out the strategic directions and open problems for the broader area of structured machine learning for the next 10 years.

Keywords Inductive logic programming · Relational learning · Statistical relational learning · Structured machine learning

Editors: Hendrik Blockeel, Jude Shavlik.

T.G. Dietterich · P. Tadepalli (✉)
Oregon State University, Corvallis, OR, USA
e-mail: tadepall@eecs.orst.edu
url: http://www.eecs.orst.edu/~tadepall

T.G. Dietterich
e-mail: tgd@eecs.orst.edu
url: http://www.eecs.orst.edu/~tgd

P. Domingos
University of Washington, Seattle, WA, USA
e-mail: pedrod@cs.washington.edu
url: http://www.cs.washington.edu/homes/pedrod

L. Getoor
University of Maryland, College Park, MD, USA
e-mail: getoor@cs.umd.edu
url: http://www.cs.umd.edu/~getoor

S. Muggleton
Imperial College, London, UK
e-mail: shm@doc.ic.ac.uk
url: http://www.doc.ic.ac.uk/~shm

1 Introduction

Structured machine learning refers to learning structured hypotheses from data with rich internal structure usually in the form of one or more relations. In general, the data might include structured inputs as well as outputs, parts of which may be uncertain, noisy, or missing. Applications of these methods include a variety of tasks such as learning to parse and translate sentences (Liang et al. 2006), predicting the pharmacological properties of molecules (Finn et al. 1998), and interpreting visual scenes (Fern and Givan 2006). While traditionally studied as part of inductive logic programming (ILP), there has been a surge of interest in structured machine learning in recent years as exemplified by several specialized workshops and at least three edited volumes of papers (Bakir et al. 2007; Getoor and Taskar 2007; De Raedt et al. 2008). By addressing learning in the context of rich representations that allow sophisticated inference, structured machine learning has the best chance of providing the tools for building integrated AI systems.

Machine learning research involved structured representations from the beginning. The early work of Evans in recognizing analogies (Evans 1968), the work of Winston on learning structured classification (Winston 1975), the research on learning macro-operators for planning (Fikes et al. 1972), and the cognitive science work of Anzai and Simon on learning to solve problems (Anzai and Simon 1979) are some well-known examples of structured learning. Perhaps more importantly, the work of Plotkin on inductive generalization of logic formulae (Plotkin 1969), and the work of Shapiro on automatic debugging (Shapiro 1983) proved to be extremely influential for the later developments in inductive logic programming (ILP).

Work on learning logical representations continued during the eighties under the umbrellas of inductive learning and explanation-based learning. In inductive learning, one seeks a logical theory that entails all positive examples and does not entail the negative examples (Dietterich and Michalski 1985). Explanation-based learning, on the other hand is deductive, in that the system already has background knowledge that entails all the positive examples. However, the background knowledge is in an intractable form and the goal is to find an efficient specialization which is sufficient to entail all positive examples (Mitchell et al. 1986; DeJong and Mooney 1986). Inductive logic programming (ILP) generalizes the inductive and the deductive approaches by aiming to find a logical theory that entails the positive examples (and not the negative examples) when conjoined with the background knowledge (Muggleton and Feng 1990; Quinlan 1990). See Table 1 for a specification of the entailment relationships implemented by these approaches, where B is the background knowledge, ξ^+ and ξ^- are positive and negative examples respectively, and h is an hypothesis selected from the hypothesis space $\mathcal{H}$. The series of workshops and conferences on ILP since 1991 have enabled in-depth exploration of learning in logical and relational representations. A number of ILP systems were developed, and several applications were demonstrated (for example, see Muggleton and De Raedt 1994; Lavrač and Džeroski 1994). Definite-clause Horn programs were the hypotheses of choice, although generalizations to more expressive languages were also considered.

It is instructive to trace the evolution of this field through a concrete example. Consider the task of part-of-speech (POS) tagging in natural language processing. The problem is to assign correct parts of speech to the words in a sentence, e.g., the words in the sentence "John went to the bank," would be given the tag sequence "np vbd to dt nn." While the word "bank" could be a noun or a verb and has multiple meanings in each case, the context surrounding the word in the first sentence suggests that it is a noun. Cussens describes an application of ILP to learn POS tagging (Cussens 1997). Each word is initially assigned to

Table 1 The entailment relationships implemented by three different paradigms of learning logical representations. B is the background knowledge, ξ^+ and ξ^- are respectively positive and negative examples, and $h \in \mathcal{H}$ is an hypothesis

Learning Paradigm	Specification: Find an hypothesis $h \in \mathcal{H}$ such that
Inductive Learning	$h \models \xi^+$ and $h \not\models \xi^-$
Explanation-based Learning	$B \models h$ and $h \models \xi^+$
Inductive Logic Programming	$B \wedge h \models \xi^+$ and $B \wedge h \not\models \xi^-$

a set of potential POS tags using the dictionary. Elimination rules are then used to remove incorrect POS tags using the properties of the surrounding context of POS tags. An ILP system called Progol is used to learn the elimination rules from a training corpus of tagged text (Muggleton 1995). For example, one could learn a rule that says that "bank" cannot be a verb if it follows a determiner ("dt") from the above positive example and a few negative examples. Progol uses background knowledge in the form of an approximate definite clause grammar, e.g., definitions of grammatic variables like noun phrase and verb phrase in terms of more primitive variables. It searches for higher level elimination rules which can be used along with the background knowledge to eliminate the incorrect POS tags from the training data.

Given the complexity of the tagging task, not all incorrect POS tags can be eliminated by such deterministic rules. Cussens uses a hybrid approach where after the elimination step, a dictionary is used to select the most frequent of the remaining POS tags for each word. The above problem motivates an approach to dealing with reasoning under uncertainty. One way to formulate the problem is to maximize the conditional probability of all tags in the sentence $Y_{1:n}$, given the words $X_{1:n}$. However, the above approach deals with the influence of the word X_i and the influence of the labels of other words, i.e., the context $Y_{1:i-1,i+1:n}$, on the current label Y_i separately. The context rules are deterministic and are applied before the word-based probability is taken into account. A more principled approach uses all the available information to make the strongest possible inference. For example, applying the word-based probabilities might in turn help eliminate candidate tags for the surrounding words.

The need to handle uncertainty in a principled manner led many to consider ways to incorporate probabilities into logical and relational representations. Several different formalisms were developed by combining the insights gained from directed and undirected graphical models, logic, and relational data bases (Getoor and Taskar 2007). In directed statistical relational models such as probabilistic relational models (PRMs) (Friedman et al. 1999), the background knowledge is represented as a relational schema describing objects and their relationships and a collection of conditional probabilities described over objects. The conditional probabilities allow the attributes of objects (and the existence of objects) to depend on attributes of other, related, objects, and are described in a compact, parameterized manner. Given a collection of objects and their observed relationships, the joint probability of any complete assignment of attributes, $P(X, Y)$ is simply the product of the appropriate entries in the associated conditional probabilities models. These directed models have a natural generative semantics, which can capture dependencies among constituents of the sentence.

In undirected statistical relational models such as Relational Markov Networks (RMNs) (Taskar et al. 2002) and Markov Logic (ML) (Richardson and Domingos 2006), the background knowledge is represented by a set of parameterized relational or first order formulas Φ_i with weights w_i. The formulas might involve joint predicates over both the structured inputs X and structured outputs Y of the prediction task. The weight represents the value of

satisfying the formula. The conditional probability $P(Y|X)$ is given by a log-linear model $\frac{1}{Z_x} e^{\sum_i w_i n_i(X,Y)}$, where $n_i(X, Y)$ represents the number of ground formulas of Φ_i that are true for X, Y, and Z_x is the normalizing factor. In either the directed or undirected case, finding the most probable Y for any X corresponds to computing

$$\operatorname*{argmax}_Y P(Y|X).$$

In POS tagging using ML, the background knowledge would include weighted rules, which might be softened versions of rules used by Cussens or even more expressive ones. For example, a formula Φ_i might say that every sentence should have a verb. Given a sentence x, all the tag sequences that satisfy the formula Φ_i, i.e., those that contain a word which is tagged as a verb, will have a score w_i added to them and the others do not. The tag sequence with the highest score is given out as the answer.

This immediately points to two problems. First, it appears that one would have to search all exponentially many tag sequences to find the argmax during performance. Second, how should such a beast be trained? The argmax problem can be solved by approximate weighted MaxSAT solvers. It also reduces to finding the maximum a posteriori (MAP) hypothesis in a conditional random field (CRF) whose nodes correspond to all possible groundings of the predicates in the ML formulas (Lafferty et al. 2001). However, since exact inference is highly intractable, one would typically use approximate methods such as simulated annealing.

MLN training has two parts: structure learning, i.e., learning of formulas, and weight learning. ILP methods have been adapted to the structure learning problem with some success (Kok and Domingos 2005). There are a number of discriminative methods for weight learning including the voted perceptron algorithm, and some second order methods (Bertsekas 1999; Nocedal and Wright 1999).

Typically discriminative methods perform better than generative methods since modeling the generative process is often much more difficult than making relevant discriminations for the task at hand. This observation led to the study of margin maximization methods and the formulation of the weight learning problem as quadratic programming problem in the support vector machine (SVM) framework. One seeks to maximize the margin between the optimal y_i and all other y's summed over all training examples (x_i, y_i). Unfortunately this results in exponentially many constraints corresponding to all other possible y's for the same x. In POS-tagging, this corresponds to the set of all incorrect tag sequences. One solution to this problem, pursued in Maximum Margin Markov networks, is to reformulate the optimization problem into a different polynomially sized one, by taking advantage of the structure of the underlying Markov network and then solve it (Taskar et al. 2003b). Another approach, pursued in SVMStruct, is to add constraints incrementally and keep solving the optimization problem until the weights converge (Tsochantaridis et al. 2005). Recently there have been some approximate versions of this approach based on online learning (Crammer et al. 2006) and gradient boosting (Parker et al. 2006, 2007) that are found to be quite effective.

In summary, there has been an explosion of new research in structured machine learning in recent years with a number of new problems and approaches for solving them. There seems to be a greater need than ever to develop a coherent vision of these emerging research areas and chart out some strategic directions for future research. The rest of this article gives the emerging trends and promising research topics in this area from the point of view of the individual contributors and concludes with a summary.

2 The next ten years of ILP
Stephen Muggleton

One of the characteristic features of the development of ILP to date has been the intertwined advancement of theory, implementations and applications. Challenging applications such as those found in areas of scientific discovery (Muggleton 2006) often demand fundamental advances in implementations and theory. Recent examples of such application-led development of implementations and theory include the tight integration of abductive and inductive inference in Systems Biology applications (Tamaddoni-Nezhad et al. 2006, 2007), the development of approaches for integrating SVMs and ILP for applications in drug discovery (Amini et al. 2007) and the development of novel ILP frameworks for mathematical discovery (Colton and Muggleton 2006).

2.1 ILP—some future directions

Automatic bias-revision: The last few years have seen the emergence within ILP of systems which use feedback from the evaluation of sections of the hypothesis space (DiMaio and Shavlik 2004), or related hypothesis spaces (Reid 2004), to estimate promising areas for extending the search. These approaches can be viewed as learning a meta-logical theory over the space of theories and we refer to them as *automatic bias-revision*. In the longer-run a theoretical framework needs to be developed which allows us to reason about possible choices for representing and effectively deploying such meta-theories. Some form of logic seems the obvious choice for such meta-theories.

Learning equational theories: Authors such as Milch and Russell (2006) have recognized that in certain learning applications it is necessary to reason about the identity of objects. For instance, when provided with images of vehicles from multiple viewpoints within road traffic data one may need to hypothesize that vehicles found in separate images represent the same physical object. Within a logic-based representation a natural way of representing such hypotheses is to employ an equational theory in which we can explicitly state that one object identifier is equivalent to another. We can then use the standard axioms of equality to allow us to reason about the transitive and symmetric consequences of such hypotheses. Equational reasoning has not been widely employed in ILP systems to date, but would be a natural extension of the logic-based frameworks which have been used.

Object invention: Going beyond issues related to object identity, it is a characteristic of many scientific domains that we need to posit the existence of hidden objects in order to achieve compact hypotheses which explain empirical observations. We will refer to this process as *object invention*. For instance, object invention is required when unknown enzymes produce observable effects related to a given metabolic network. It would be natural to integrate any theory of object invention within the context of systems which use equational reasoning. It is worth noting that object invention was at the heart of many of the great conceptual advances in Science (e.g., atoms, electrons and genes) and Mathematics (e.g., 0, π and i).

Incremental theory revision: The simpler approaches to ILP assume complete and correct background knowledge. Theory revision is an ILP setting in which we assume background knowledge to be both incomplete and incorrect. Localized alterations of clauses in the background knowledge are then used to correct and complete predictions on the training data.

Although in the past there was considerable effort in development of theory revision systems (Wrobel 1995), the lack of applications with substantial codified background knowledge meant that such systems were not widely deployed. The situation has changed in recent years with the availability of large-scale resources of background knowledge in areas such as biology (Muggleton 2005). As a result there is renewed interest in theory revision systems, especially in cases involving probabilistic logic representations (Paes et al. 2005). Owing to the rapid growth of public databases in the sciences there is an increasing need for development of efficient theory revision ILP systems.

Logical experiment design: One of the key issues in scientific applications is the design of experiments to test hypotheses under exploration. Within Machine Learning the area associated with experimental validation of hypotheses is known as active learning. An active learning system based on ILP, such as that employed in the Robot Scientist (Bryant et al. 2001; King et al. 2004) chooses experiments whose outcome is expected to rapidly reduce the hypothesis space. However, the approach assumes a fixed and finite experiment space. Within an ILP framework we should explore whether it is possible to use background knowledge to devise logical descriptions of experiments to be carried out. If successful, such an approach would have immense benefit in laboratory settings.

2.2 Probabilistic ILP—what is needed

2.2.1 Theory

Statistical Relational Learning (SRL) (Getoor and Taskar 2007), is the sub-area of Machine Learning which deals with the empirical construction of inductive hypotheses which contain both a relational and a statistically estimated component. By contrast, the related field of probabilistic inductive logic programming (PILP) (De Raedt and Kersting 2004), is the sub-area of machine learning in which the examples, hypotheses and background knowledge are represented in the form of a probabilistic logic. After six years of development the field of Probabilistic ILP (PILP) or Statistical Relational Learning (SRL) is in a weaker state with respect to its theoretical foundations than ILP was at a similar point in its development. Several areas need urgent further development to rectify this situation.

Clarify logical/probabilistic settings: The logical/probabilistic assumptions made are still often unclear in SRL papers. The situation is better in PILP (De Raedt and Kersting 2004) in which clear delineations of the roles of background knowledge hypotheses and examples are made. However, the role of probabilistic information is still in flux. For instance, the issue of whether background knowledge is necessarily probabilistic varies from one system to another. Similarly the question of whether examples should be labelled with probabilities is again unsettled. Overall more work needs to be put into characterizing the various settings in which PILP/SRL systems can operate.

Learnability model and results: Considerable work has been carried out on the expressivity relationships between various PILP/SRL representations (e.g., Puech and Muggleton 2003) However, the author knows of no published attempt to develop a learnability model for PILP/SRL. This puts the area in a weak position in theoretical terms, and is especially of concern given the fact that the use of non-iid data makes it difficult to transform PAC-style learning approaches for these purposes.

2.2.2 *Implementations*

There is a real need for more standard and efficient benchtest systems to be made available in the PILP/SRL area. PRISM (Sato 2005), Alchemy (Domingos et al. 2006), and FAM (Cussens 2001) are important and useful examples of such systems, though more needs to be done in benchtesting these systems against each other.

2.2.3 *Experimental applications*

Despite the prominence of the area, applications of PILP/SRL are still narrow, with overuse of examples associated with citation analysis. The following is a list of potentially exciting application areas for PILP/SRL.

- *Scientific and mathematical discovery problems:* A large number of datasets are available in this area from previous applications of ILP.
- *Natural language learning:* Representations which mix logical and statistical elements fit very naturally with grammar learning tasks in natural language.
- *Hardware/software verification:* The use of logical and probabilistic representations should be a powerful combination in both hardware and software verification.

2.3 Conclusions

ILP has had a rich and exciting history of development. The next ten years in ILP and the related fields of PILP/SRL should lead to a deepening and strengthening of achievements to date. We argue that developments in ILP should be application-led, with efficient widely-distributed benchtest systems developed and tested within a well-defined theoretical framework.

3 Structured machine learning: past, present and future
Lise Getoor

I will structure my summary as a sort of Dickensian story of research in structured machine learning, with a somewhat biased focus on work from the statistical relational learning (SRL) community.

3.1 The past: alphabet soup

In the past ten years, there has been a huge amount of research in methods which combine structured logical representations with uncertainty and probabilistic representations. The syntax and semantics for the logical representations has varied, including rule-based and frame-based approaches, and methods based on programs or grammars. The syntax and semantics for the probabilistic representations have also varied, although a large proportion of them are based on graphical models, either directed graphical models such as Bayesian networks, or undirected graphical models such as Markov networks.

During this time, there has been a proliferation on representations, beginning with the early work on Probabilistic Horn Abduction (PHA) (Poole 1993) and Knowledge-Based Model Construction (KBMC) (Wellman et al. 1992), and continuing with the work on Stochastic Logic Programs (SLPs) (Muggleton 1996), Bayesian Logic Programs (BLPs) (Kersting et al. 2000), Probabilistic Relational Models (PRMs) (Koller and Pfeffer 1998;

Friedman et al. 1999; Getoor et al. 2001a), Relational Bayesian Networks (RBNs) (Jaeger 1997), Relational Markov Networks (RMNs) (Taskar et al. 2002), Markov Logic networks (MLNs) (Richardson and Domingos 2006), Relational Dependency Networks (RDNs) (Neville and Jensen 2003, 2007), Probabilistic Entity Relationship Models (PERs) (Heckerman et al. 2004), Prism (Sato and Kameya 1997), CLP-BN (Costa et al. 2003), Bayesian Logic (BLOG) (Milch et al. 2004), IBAL (Pfeffer 2001), and more.

Both the logical syntax and semantics and the probabilistic syntax and semantics of the proposed systems vary, and just about every combination has been tried. Logic representations based on Horn clauses, frame-based systems, constraint-based systems and first-order logic have been proposed. Probabilistic representations have been proposed based on directed graphical models (aka Bayesian networks), undirected graphical models (aka Markov networks), stochastic context free grammars, and functional programming languages.

Interestingly, despite the plethora of different representations, some common issues and themes have emerged. The issues include 1) dealing with many-many and many-one relationship requires some form of aggregation or combining rules; 2) dealing with structural uncertainty requires some effective way of representing distributions over the (large) number of possible logical interpretations; 3) dealing with open-world semantics requires some way of introducing new, generic, constants; 4) dealing with a mix of observed and unobserved data requires some method for making use of both during inference and learning; and 5) dealing with background knowledge requires some way of effectively making use of logical knowledge in the form of relational schema and/or ontologies to constraint or bias the structure of the probabilistic model. Each proposed representation may deal with them slightly differently, and often times the proposed solution translates relatively directly from one representation to another, but this seems like an area where good progress has been made.

3.2 The present: tasks

Currently, within the SRL community, there has been a (healthy, in my opinion) move from the focus on representations to a focus on the types of inference and learning tasks and applications that can be solved using SRL techniques. There has been cross-fertilization with research in diverse areas such as natural language processing, computer vision, social network analysis, bioinformatics, and the semantic web. In many of these areas there has been a split between 'statistical' approaches and 'logical' or 'semantic' approaches, and there is a perceived need for bridging the two.

One extremely common inference and learning task in all of these application areas is collective classification. In structured data, the classification of linked objects is usually not independent; because they are linked they are more likely to have correlated labels (autocorrelation), or because they have similar labels, they are more likely to be linked (homophily). There is a long tradition of work in methods which optimize the joint labels for a network of random variables, beginning with work on relaxation labeling in computer vision (Rosenfeld et al. 1976), which has been applied to web page classification by Chakrabarti et al. (1998). Neville and Jensen proposed a simple iterative classification algorithm for collective classification (Neville and Jensen 2000), and later studied relational dependency networks for the tasks (Neville and Jensen 2003). Getoor et al. studied collective classification in probabilistic relational models (Getoor et al. 2001b), and Taskar et al. studied collective classification in relational Markov networks (Taskar et al. 2001). There has been much follow on work, e.g., Lu and Getoor (2007) and Macskassy and Provost (2007), to name a few.

Another extremely useful inference and learning task is entity resolution. Entity resolution is the problem of determining, from a collection of multiple, noisy, interrelated references, the true underlying set of entities in the domain, and the mapping from the references to the entities. The problem comes up naturally in many domains including natural language processing (co-reference resolution), computer vision (the correspondence problem) and data bases (the deduplication or record linkage problem). Pasula et al. (2002) first studied the problem in a generic SRL setting. Since then there has been much work including work on relational clustering for entity resolution (Bhattacharya and Getoor 2004) and a variety of probabilistic approaches to entity resolution in relational domains.

Another important inference and learning task is link prediction. Link prediction involves predicting whether or not a relationship exists between two entities. At first glance, this appears to be just another collective classification task, but because of the large space of potential related objects, and the need to be able to generalize and make predictions for new unseen objects, different models are often appropriate. Getoor et al. (2002) introduced two approaches in the context of probabilistic relational models, and the general link prediction problem has been studied by many others (Liben-Nowell and Kleinberg 2003; Kubica et al. 2002; Taskar et al. 2003a).

The final inference and learning task I will mention is group detection. Group detection in relational domains, especially graph data, has received a great deal of attention. In some cases, the graphs are simply similarity graphs where edges between nodes are weighted by the similarity of the nodes. For these, graph cut algorithms work well, as do graph-kernel approaches (Gärtner 2003). There have also been approaches developed for richer relational domains, such as the work on clustering in PRMs (Taskar et al. 2001).

3.3 The future: open problems and connections to other research areas

One of the biggest open challenges, and one that is likely to keep this community employed for quite some time, is how to build inference and learning algorithms that can jointly solve all of the above tasks (and probably others) in a scalable and reliable manner. Scalability means having learning and inferences algorithms that can figure out how to bring in and reason about only the information necessary to answer the query at hand; this is especially important in the context of decision making, where more information may provide more precise probability estimates, but may be unlikely to change the best action choice. Reliability means having algorithms which can quantify in some manner the confidence and sensitivity of the results. Even in the case of focused inferences for entity resolution and link prediction, it is important to be able to qualify the confidence in whether two references refer to the same underlying individual or whether the link between two individuals exists.

There are a number of other research areas that may help towards making structured machine learning algorithms scalable and reliable. One area that may help with scalability is the recent and growing work in the area of probabilistic databases. A tighter connection between the work in structured machine learning and probabilistic databases is likely to result in systems which are more efficient and practical. Another research area that will help make the output of structured machine learning algorithms more reliable is visualization support which allows users to inspect the model being learned and inferred. Ideally these tools will allow for easy inspection and quantification of the uncertainty associated with various conclusions, and allow users to easily modify and add constraints to the model.

In order to be practically relevant, it is important for researchers in structured machine learning to continue to apply their work to practical real-world problems. This has already proven useful within domains such as natural language processing, computer vision and

bioinformatics, and we should continue to seek out these "killer applications." Personal information management is one which I find very compelling; this comes hand-in-hand with many potential privacy pitfalls. Structured machine learning methods may also be able to provide useful insight into privacy and information disclosure in relational domains.

Long-term, dealing with the dynamic nature of systems is important. In the real world, we often must deal with shifting sands, non-stationary distributions, and other complications that mean that our clean theoretical models are only convenient approximations. Nonetheless, we want systems that degrade gracefully or in some other manner exhibit "elaboration-tolerance." Structured machine learning is a step in that direction.

4 Open problems in structured machine learning

Thomas G. Dietterich

There is one overarching problem for structured machine learning: developing scalable learning and reasoning methods. I see three subproblems within this overall problem.

First, many of the initial methods for structured learning were batch methods based on extensions of logistic regression (e.g., conditional random fields) and support vector machines (e.g., SVMStruct, Maximum Margin Markov networks). One important direction, which is already very active, is to develop online learning methods for structural problems. Collins (2002) initiated this line of research with his voted perceptron methods for training CRFs. More recently, Crammer and his colleagues (Crammer et al. 2006) have applied the Passive-Aggressive family of online "margin-infused" algorithms in this area. An added benefit of online, incremental learning methods is that they have the potential to deal with nonstationary environments.

A second major challenge is to develop methods for automatically decomposing large structured reasoning problems into more tractable subproblems. As we tackle larger and larger structured and relational learning problems, the cost of inference in these problems comes to dominate learning time and makes performance very slow. This trend can only worsen as we consider large integrated systems for relational learning and reasoning. Hence, we need to find ways to reduce the cost of inference both at learning time and at run time. One promising direction is to identify subproblems for which efficient inference is possible, as in the work of Duchi et al. (2007). A related strategy for efficient learning is the piecewise training method of Sutton and McCallum (2007).

A third open problem is to integrate learning into the reasoning process itself. In many structured learning problems, the primary inference task is to compute

$$\operatorname*{argmax}_{Y} \Phi(X, Y; \Theta)$$

where X is the input structure, Y is the output structure, and Θ are the parameters of the scoring function Φ. In some settings, such as sequence labeling and context-free grammar parsing, polynomial-time dynamic programming algorithms exist to perform this computation. However, even in those cases, we need much faster methods to be able to handle large problems. For example, in label sequence learning, if there are K possible labels and the sequences are of length T, then the Viterbi algorithm requires $O(K^2T)$ time. For some problems (e.g., semantic role labeling, text-to-speech mapping), K is on the order of 100, so this scales as $T \times 10^4$, which is far too expensive to include within the inner loop of a learning procedure.

One promising approach is to apply some form of beam search or greedy search to compute the argmax. For example, we could process X and Y from left-to-right and compute a partial score $\Phi_{1:t}(X_{1:t}, Y_{1:t})$ over the first t items in the sequence. The hope is that the features describing $X_{1:t}$ and $Y_{1:t}$ are sufficient to allow a learning algorithm to score the correct output sequence $Y^*_{1:t}$ within the top B candidates, where B is the width of the beam search. This may require additional feature engineering, but it has the potential to reduce the cost of inference to $O(BKT)$, where $B < K$ is the beam width. Collins and Roark (2004) applied this beam-search formulation in combination with their perceptron training approach. They performed a perceptron update as soon as the right partial answer $Y^*_{1:t}$ failed to score in the top B candidates. Related results have been obtained by Daumé and Marcu (2005), Daumé et al. (2007) and by Xu and Fern (2007), Xu et al. (2007). The most extreme version of this is where $B = 1$ so that beam search is reduced to a simple greedy algorithm. Culotta et al. describe a co-reference resolution problem in which the Y space consists of all partitions of a set of "mentions" (i.e., referring expressions) into equivalence classes. They employ a bottom-up agglomerative clustering algorithm to solve this problem, and train it with a variant of Crammer's Passive Aggressive algorithm (Culotta et al. 2007) that gives a major improvement in accuracy on this task.

This existing work shows that it is possible to integrate learning into the reasoning process. However, we still are only just beginning to understand the theoretical basis for the success of these methods.

4.1 Applications

Let me briefly describe two applications for structured machine learning.

The first application problem concerns predicting the distribution of species. In the single-species problem, we are given training data of the form $(\mathbf{x}, y)$ where $\mathbf{x}$ is a feature vector describing a particular site and $y = 1$ is true if the species is present at that site and 0 otherwise. This is a standard supervised learning problem. It becomes a structured problem if the sites are geolocated and we wish to capture the fact that nearby sites may have similar presence/absence of the species. Another way in which the problem can become structured is if we replace y by a vector $\mathbf{y}$ of presence/absence indicators for K different species. For example, Leathwick et al. (2005) describe a problem in which $\mathbf{y}$ provides presence/absence information for 15 different fish species. We expect that the presence of different species will not be independent but correlated because of competition, predator-prey relationships, and so on. These inter-species relationships might be pairwise, or species might organize into communities. In other problems (e.g., plants, arthropods), K may be 4000.

A second application problem arises in integrated intelligent systems such as the CALO (Computer Assistant that Learns and Organizes) system (Mark and Perrault 2007). Such systems need to mix human-authored and machine-learned probabilistic knowledge in order to maintain an up-to-date model of the beliefs of the system. This combines state estimation and activity recognition. Some aspects of the state (e.g., files, email messages, folders) are easy to observe. Others (e.g., the user's projects, responsibilities, action items, commitments, goals, and intentions) must be inferred. Figure 1 shows the project-oriented fragment of the CALO relational model. Solid lines indicate objects and relationships that are observed; dashed lines indicate objects and relations that must be inferred. Note, for example, that although the set of projects is observed, the relationships between files and projects, folders and projects, people and projects, etc. must all be inferred. The set of people must be inferred from the names and email addresses in email messages.

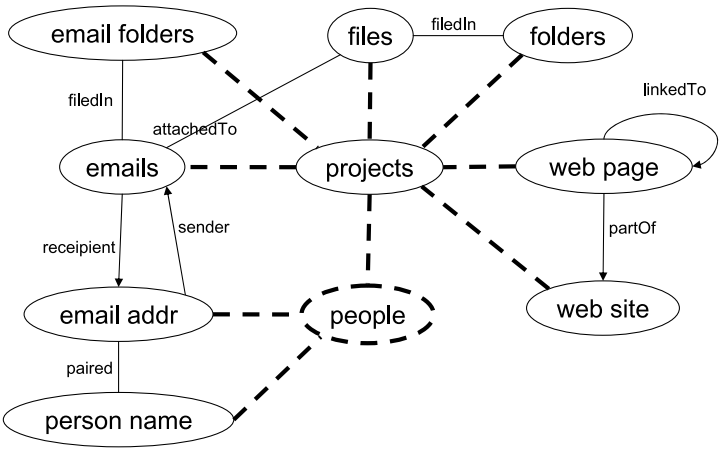

Fig. 1 Relational model for desktop computing in CALO

4.2 Challenges for Markov Logic

Markov Logic (Richardson and Domingos 2006) provides a general-purpose system for relational probabilistic modeling, learning, and inference. Nonetheless, there are several things that are currently difficult or impossible to represent in Markov Logic, so they provide good challenge problems for future research. I describe three such problems here.

First, consider a computer vision problem where we wish to express a probabilistic relationship between the orientation of a street and the orientation of the sidewalk next to the street. We would like to say something like "sidewalks should be parallel to streets." A natural way to do this would be to define an orientation vector (walk.v or street.v) that captures the principal orientation of the object in some appropriate coordinate system. Then two objects are parallel if the dot product of their orientation vectors is 1. We would then like to define a Markov network potential function that depends on this dot product: $\Phi(\text{walk}.v \cdot \text{street}.v)$. This is impossible in current Markov Logic systems, because each potential function is represented by a set of weighted clauses, and the weight of each clause is a (learned) constant. In effect, this represents a tabular potential function analogous to a conditional probability table in Bayesian networks, but we need a continuous potential function instead. The root problem is that we need to move beyond weighted first-order logic to more general weighted algebraic constraint systems.

Second, consider the co-reference resolution work of Culotta and McCallum mentioned above. Earlier I focused on the way in which they integrate learning with bottom-up agglomerative clustering for inference. But another key aspect of this work is their representation. Much of the success of their system comes from their use of features that describe global properties of each cluster. For example, a cluster that includes three mentions "He", "She", and "Vanessa Williams" is a bad cluster, because the mentions in the cluster do not agree on gender. How can these kinds of features be compactly represented in Markov Logic?

Third, in many applications, there are cases in which the Markov network potential function should depend on the number of objects in a set. For example, in the CALO relational model, a file can be associated with multiple projects, but most likely only 1, 2, or 3 and not 20, 30, or 100. It is possible to employ conjunctions that test whether a file belongs to at

least K projects and then assign a penalty to this. For example, $K = 3$ can be represented as

$$\exists f, p1, p2, p3 ProjectOf(f, p1) \wedge ProjectOf(f, p2) \wedge ProjectOf(f, p3) \\ \wedge p1 \neq p2 \wedge p2 \neq p3 \wedge p1 \neq p3.$$

However, this must be instantiated for every triple of projects, which is very inefficient. Instead, we would like to refer directly to the cardinality of the set of projects associated with a file:

$$ProjectCount(f) = |\{p : ProjectOf(f, p)\}|.$$

Then we could define potentials for $ProjectCount(f) = 1$, $ProjectCount(f) = 2$, and so on. This special case can be implemented very efficiently. The larger challenge is to develop a language that can express these kinds of things and yet can be compiled into efficient implementations.

4.3 Possible dissertation topics

Here are three possible dissertation topics that instantiate the challenges raised above:

- Automated decomposition methods for large Markov Logic systems. This would present methods for automatically decomposing large Markov Logic systems to support efficient learning and reasoning.
- Theory and practice of learning to reason. This would develop the theory to explain when it is and is not possible to incorporate learning into the inference process.
- Markov Constraint Logic. This would extend Markov Logic to support real-valued potential functions.

5 Structured machine learning: ten problems for the next ten years
Pedro Domingos

5.1 Statistical predicate invention

Predicate invention in ILP and hidden variable discovery in statistical learning are really two faces of the same problem. Researchers in both communities generally agree that this is a key (if not the key) problem for machine learning. Without predicate invention, learning will always be shallow. In essence, every word in the dictionary is an invented predicate, with many layers of invention between it and the sensory percepts on which it is ultimately based. Unfortunately, progress to date has been limited. The consensus seems to be that the problem is just too hard, and it is not clear what to do about it. However, combining predicate invention and latent variable discovery into the single problem of statistical predicate invention may lead to new breakthroughs. (One is reminded of Eisenhower's saying: "If you can't solve a problem, magnify it.") Considering statistical and logical aspects simultaneously gives us both more opportunities and more constraints to work with. It is also a natural continuation of the statistical relational learning agenda. For some preliminary ideas, see Kok and Domingos (2007).

5.2 Generalizing across domains

Machine learning has traditionally been defined as generalizing across tasks from the same domain, and in the last few decades we have learned to do this quite successfully. However, the glaring difference between machine learners and people is that people can generalize across domains with great ease. For example, Wall Street hires lots of physicists who know nothing about finance, but they know a lot about particle physics and the math it requires, and somehow this transfers quite well to pricing options and predicting the stock market. Machine learners can do nothing of that kind. If the predicates describing two domains are different, there is just nothing the learner can do in the new domain given what it learned in the old one. The key insight that seems to be missing is that domains have structural similarities, and we can detect them and exploit them. For example, two domains might be described by the same formula(s), but over different predicates, and having learned the formula in one domain it should be easier to rediscover it in another. This seems like an ideal challenge for relational learning, since in some sense it is "extreme relational learning": we are not just using relations to generalize, we are using relations between relations. DARPA has recently started a project in this area, but so far we have only scratched the surface. (For a good example, see Mihalkova et al. 2007.)

5.3 Learning many levels of structure

So far, in statistical relational learning (SRL) we have developed algorithms for learning from structured inputs and structured outputs, but not for learning structured internal representations. In both ILP and statistical learning, models typically have only two levels of structure. For example, in support vector machines the two levels are the kernel and the linear combination, and in ILP the two levels are the clauses and their conjunction. While two levels are in principle sufficient to represent any function of interest, they are an extremely inefficient way to represent most functions. By having many levels and reusing structure we can often obtain representations that are exponentially more compact. For example, a BDD (Boolean decision diagram) can represent parity with a linear number of operations, while clausal form requires an exponential number. This compactness should also be good for learning, but nobody really knows how to learn models with many levels. The human brain has many layers of neurons, but backpropagation seldom works with more than a few. Hinton and others have begun to work on learning "deep networks," (Hinton et al. 1993), but they are not very deep yet, and they are only for unstructured data. Recursive random fields, proposed in our IJCAI-2007 paper (Lowd and Domingos 2007), are a potentially "deep" SRL representation, but learning them suffers from the limitations of backpropagation. Clearly this is an area where there is much to be discovered, and where progress is essential if we are to ever achieve something resembling human learning.

5.4 Deep combination of learning and inference

Inference is crucial in structured learning, but research on the two has been largely separate to date. This has led to a paradoxical state of affairs where we spend a lot of data and CPU time learning powerful models, but then we have to do approximate inference over them, losing some (possibly much) of that power. Learners need biases and inference needs to be efficient, so efficient inference should be the bias. We should design our learners from scratch to learn the most powerful models they can, subject to the constraint that inference over them should always be efficient (ideally real-time). For example, in the paper

"Naive Bayes Models for Probability Estimation," we learned models that were as accurate as Bayesian networks, but where inference was always linear-time, as opposed to worst-case exponential (Lowd and Domingos 2005). In SRL efficient inference is even more important, and there is even more to be gained by making it a goal of learning.

5.5 Learning to Map between representations

An application area where structured learning can have a lot of impact is representation mapping. Three major problems in this area are entity resolution (matching objects), schema matching (matching predicates) and ontology alignment (matching concepts). We have algorithms for solving each of these problems separately, assuming the others have already been solved. But in most real applications they are all present simultaneously, and none of the "one piece" algorithms work. This is a problem of great practical significance because integration is where organizations spend most of their IT budget, and without solving it, the "automated Web" (Web services, Semantic Web, etc.) can never really take off. It seems like an ideal problem for joint inference: if two objects are the same, then perhaps the fields they appear in are the same, and in turn the concepts containing those fields may be the same, and vice-versa. And learning for joint inference is what SRL is all about, so this could be a "killer app." Beyond one-to-one mapping lies the deeper challenge of learning to convert from one representation of a problem in logic to another. Humans can recognize when two sets of formulas are essentially saying the same thing, even when they are not logically equivalent. AI systems should be able to do the same.

5.6 Learning in the large

Structured learning is most likely to pay off in large domains, because in small ones it is often not too difficult to hand-engineer a "good enough" set of propositional features. So far, for the most part, we have worked on micro-problems (e.g., identifying promoter regions in DNA); our focus should shift increasingly to macro-problems (e.g., modeling the entire metabolic network in a cell). We need to learn "in the large," and this does not just mean large datasets. It has many dimensions: learning in rich domains with many interrelated concepts; learning with a lot of knowledge, a lot of data, or both; taking large systems and replacing the traditional pipeline architecture with joint inference and learning; learning models with trillions of parameters instead of millions; continuous, open-ended learning; etc.

5.7 Structured prediction with intractable inference

Max-margin training of structured models like HMMs and PCFGs has become popular in recent years. One of its attractive features is that, when inference is tractable, learning is also tractable. This contrasts with maximum likelihood and Bayesian methods, which remain intractable. However, most interesting AI problems involve intractable inference. How do we optimize margins when inference is approximate? How does approximate inference interact with the optimizer? Can we adapt current optimization algorithms to make them robust with respect to inference errors, or do we need to develop new ones? We need to answer these questions if max-margin methods are to break out of the narrow range of structures they can currently handle effectively.

5.8 Reinforcement learning with structured time

The Markov assumption is good for controlling the complexity of sequential decision problems, but it is also a straitjacket. In the real world, systems have memory, some interactions are fast and some are slow, and long uneventful periods alternate with bursts of activity. We need to learn at multiple time scales simultaneously, and with a rich structure of events and durations. This is more complex, but it may also help make reinforcement learning more efficient. At coarse scales, rewards are almost instantaneous, and RL is easy. At finer scales, rewards are distant, but by propagating rewards across scales, we may be able to greatly speed up learning.

5.9 Expanding SRL to statistical relational AI

We should reach out to other subfields of AI, because they have the same problems we do: they have logical and statistical approaches, each solves only a part of the problem, and what is really needed is a combination of the two. We want to apply learning to larger and larger pieces of a complete AI system. For example, natural language processing involves a large number of subtasks (parsing, coreference resolution, word sense disambiguation, semantic role labeling, etc.). So far, learning has been applied mostly to each one in isolation, ignoring their interactions. We need to drive towards a solution to the complete problem. The same applies to robotics and vision, and other fields. We need to avoid falling into local optima in our research: once a problem is solved "80/20," we should move on to the next larger one that includes it, not continue to refine our solution with diminishing returns. Our natural tendency to do the latter greatly slows down the progress of research. Moreover, the best solutions to subproblems taken in isolation are often not the best ones in combination. Because of this, refining solutions to subproblems can in fact be counterproductive—digging deeper into the local optimum instead of escaping it.

5.10 Learning to debug programs

Machine learning is making inroads into other areas of computer science: systems, networking, software engineering, databases, architecture, graphics, HCI, etc. This is a great opportunity to have impact, and a great source of rich problems to drive the field. One area that seems ripe for progress is automated debugging. Debugging is extremely time-consuming, and it was one of the original applications of ILP (Shapiro 1983). However, in the early days there was no data for learning to debug, and learners could not get very far. Today we have the Internet and huge repositories of open-source code. Even better, we can leverage mass collaboration. Every time a programmer fixes a bug, we potentially have a piece of training data. If programmers let us automatically record their edits, debugging traces, compiler messages, etc., and send them to a central repository, we will soon have a large corpus of bugs and bug fixes. Of course, learning to transform buggy programs into bug-free ones is a very difficult problem, but it is also highly structured, noise-free, and the grammar is known. So this may be a "killer app" for structured learning.

6 Conclusions

Reflecting the excitement in the field, the contributors uncovered a plethora of interesting research problems in structured machine learning. In spite of the slightly different formalisms and terminologies used to describe them, there are many common concerns and

themes. Both Getoor and Dietterich emphasize the central overarching problems of scalable inference and learning in these representations. Reducing the cost of inference during both learning and performance is a central concern. While Domingos suggests learning representations that are guaranteed to be efficient for inference, Dietterich suggests learning to control inference via beam search and other divide and conquer schemes. The work by Xu and Fern suggests there are interesting expressiveness vs. learnability vs. tractability tradeoffs between these approaches (Xu and Fern 2007). Clearly there is much room for deeper theory and experimental work in this area. The themes of discovering new predicates and objects, resolving coreferences of objects, learning in rich domains with a lot of data and knowledge, and learning in non-stationary environments have arisen repeatedly.

All the contributors emphasized the importance of applications to drive the research, and there appears to be no dearth of challenging applications. Modeling geographic distribution of species, intelligent personal information systems, social network analysis, semantic web, scientific discovery, natural language processing, computer vision, hardware and software verification, and debugging were some notable applications mentioned.

Domingos advocated expanding the horizons even further to include statistical relational AI. Indeed, it seems appropriate that structured machine learning is in the best position to go beyond supervised learning and inference, and consider decision-theoretic planning and reinforcement learning in relational and first-order settings. There have been several relational extensions to planning and reinforcement learning in recent years which can be informed by the models and methods developed in structured machine learning and in turn influence these models (Tadepalli et al. 2004). Approximate versions of value iteration (Džeroski et al. 2001), policy iteration (Fern et al. 2006), and linear programming have been explored with different degrees of success (Sanner and Boutilier 2006). Deduction from factored representations of MDPs has also been tried in the form of symbolic value iteration and symbolic policy iteration, which appears promising (Kersting et al. 2004; Wang and Khardon 2007).

The authors agree that the joint workshops in logical statistical and relational learning in the past few years at Dagstuhl and other places have been instrumental in stimulating common interests and new research in this area. On the other hand, at the ILP 2007 panel discussion on structured machine learning Bernhard Pfahringer made the observation that perhaps the impact of the research in this area was somewhat muted by the division of the community into many smaller sub-communities. He proposed that an umbrella conference on structured machine learning in the near future might facilitate more interaction between the subcommunities and have a stronger impact on the broader machine learning community. We agree that there is room for one or more such conferences and hope that this article inspires more work in this area and moves us closer to the goal of realizing integrated AI.

Acknowledgements The authors thank the reviewers for their helpful comments. Thanks are also due to Ronald Bjarnason, Janardhan Rao Doppa, and Sriraam Natarajan for proof-reading the paper.

Domingos thanks Stanley Kok, Daniel Lowd, Hoifung Poon, Matt Richardson, Parag Singla, Marc Sumner, and Jue Wang for useful discussions. His research was partly supported by DARPA grant FA8750-05-2-0283, DARPA contract NBCH-D030010, NSF grant IIS-0534881, and ONR grant N00014-05-1-0313.

Muggleton's work was supported by his Royal Academy of Engineering/Microsoft Research Chair on "Automated Microfluidic Experimentation using Probabilistic Inductive Logic Programming," the BBSRC grant supporting the Centre for Integrative Systems Biology at Imperial College, Grant Reference BB/C519670/1 and the BBSRC grant on "Protein Function Prediction using Machine Learning by Enhanced Novel Support Vector Logic-based Approach," Grant Reference BB/E000940/1.

Dietterich's and Tadepalli's research was supported by DARPA contracts FA8650-06-C-7605 and FA8750-05-2-0249. Getoor's research was supported by NSF grants IIS-0308030 and IIS-0746930.

References

Amini, A., Muggleton, S. H. H. L., & Sternberg, M. (2007). A novel logic-based approach for quantitative toxicology prediction. *Journal of Chemical Informatics Modelling*, *47*(3), 998–1006. doi:0.1021/ci600223dS1549-9596(60)00223-4.

Anzai, Y., & Simon, H. A. (1979). The theory of learning by doing. *Psychological Review*, *86*, 124–140.

Bakir G. H., Hofmann T., Schölkopf B., Smola A. J., Taskar B., & Vishwanathan S. V. N. (Eds.) (2007). *Predicting structured data*. New York: MIT Press.

Bertsekas, D. (1999). *Nonlinear programming*. Belmont: Athena Scientific.

Bhattacharya, I., & Getoor, L. (2004). Iterative record linkage for cleaning and integration. In *The ACM SIGMOD workshop on research issues on data mining and knowledge discovery* (*DMKD*), Paris, France.

Bryant, C., Muggleton, S., Oliver, S., Kell, D., Reiser, P., & King, R. (2001). Combining inductive logic programming, active learning and robotics to discover the function of genes. *Electronic Transactions in Artificial Intelligence*, *5-B1*(012), 1–36.

Chakrabarti, S., Dom, B., & Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. In *International conference on management of data* (pp. 307–318).

Collins, M. (2002). Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In *Proceedings of the conference on empirical methods in natural language processing* (*EMNLP 2002*) (pp. 1–8), Morristown, NJ, USA. Association for Computational Linguistics.

Collins, M., & Roark, B. (2004). Incremental parsing with the perceptron algorithm. In *Proceedings of the association for computational linguistics* (*ACL-2004*) (pp. 111–118). Association for Computational Linguistics.

Colton, S., & Muggleton, S. (2006). Mathematical applications of inductive logic programming. *Machine Learning*, *64*, 25–64. doi:10.1007/s10994-006-8259-x.

Costa, V., Page, D., Qazi, M., & Cussens, J. (2003). CLP(BN): constraint logic programming for probabilistic knowledge. In *Proceedings of the 19th annual conference on uncertainty in artificial intelligence (UAI-03)* (pp. 517–552), San Francisco. San Mateo: Morgan Kaufmann.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online passive-aggressive algorithms. *Journal of Machine Learning Research*, *7*, 551–585.

Culotta, A., Wick, M., Hall, R., & McCallum, A. (2007). First-order probabilistic models for coreference resolution. In *HLT/NAACL* (pp. 81–88).

Cussens, J. (1997). Part-of-speech tagging using Progol. In *LNAI: Vol. 1297. Proc. of the 7th international workshop on inductive logic programming (ILP-97)* (pp. 93–108). Berlin: Springer.

Cussens, J. (2001). Parameter estimation in stochastic logic programs. *Machine Learning*, *44*(3), 245–271.

Daumé III, H., & Marcu, D. (2005). Learning as search optimization: Approximate large margin methods for structured prediction. In *Proceedings of the 22nd international conference on machine learning (ICML-2005)* (pp. 169–176). Madison: Omnipress.

Daumé III, H., Langford, J., & Marcu, D. (2007). *Search-based structured prediction* (Technical Report). University of Utah, Department of Computer Science.

De Raedt, L., & Kersting, K. (2004). Probabilistic inductive logic programming. In S. Ben-David, J. Case, & A. Maruoka (Eds.), *Lecture notes in computer science: Vol. 3244. Proceedings of the 15th international conference on algorithmic learning theory* (pp. 19–36). Berlin: Springer.

De Raedt L., Frasconi P., Kersting K., & Muggleton S. H. (Eds.) (2008). *Lecture notes in computer science. Probabilistic inductive logic programming*. Berlin: Springer.

DeJong, G., & Mooney, R. (1986). Explanation-based learning: An alternative view. *Machine Learning*, *1*, 145–176.

Dietterich, T. G., & Michalski, R. S. (1985). Discovering patterns in sequences of events. *Artificial Intelligence*, *25*(2), 187–232.

DiMaio, F., & Shavlik, J. (2004). Learning an approximation to inductive logic programming clause evaluation. In R. Camacho, R. King, & A. Srinivasan (Eds.), *Lecture notes in artificial intelligence: Vol. 3194. Proceedings of the 14th international conference on inductive logic programming* (pp. 80–96). Berlin: Springer.

Domingos, P., Kok, S., Poon, H., Richardson, M., & Singla, P. (2006). Unifying logical and statistical AI. In *Proceedings of the 21'st national conference on artificial intelligence (AAAI 2006)* (pp. 2–7). Menlo Park: AAAI Press.

Duchi, J., Tarlow, D., Elidan, G., & Koller, D. (2007). Using combinatorial optimization within max-product belief propagation. In B. Schölkopf, J. Platt, & T. Hoffman (Eds.), *Advances in neural information processing systems* (Vol. 19, pp. 369–376). Cambridge: MIT Press.

Džeroski, S., De Raedt, L., & Driessens, K. (2001). Relational reinforcement learning. *Machine Learning*, *43*, 7–52.

Evans, T. G. (1968). A program for the solution of a class of geometric-analogy intelligence-test questions. In M. Minsky (Ed.), *Semantic information processing*. Boston: MIT Press.

Fern, A., & Givan, R. (2006). Sequential inference with reliable observations: Learning to construct force-dynamic models. *Artificial Intelligence*, *170*(14–15), 1081–1122.

Fern, A., Yoon, S., & Givan, R. (2006). Approximate policy iteration with a policy language bias: Solving relational Markov decision processes. *Journal of Artificial Intelligence Research*, *25*, 75–118.

Fikes, R., Hart, P., & Nilsson, N. (1972). Learning and executing generalized robot plans. *Artificial Intelligence*, *3*, 251–288.

Finn, P., Muggleton, S., Page, D., & Srinivasan, A. (1998). Pharmacophore discovery using the Inductive Logic Programming system Progol. *Machine Learning*, *30*, 241–271.

Friedman, N., Getoor, L., Koller, D., & Pfeffer, A. (1999). Learning probabilistic relational models. In *Proceedings of the international joint conference on artificial intelligence* (pp. 1300–1307), Sweden, Stockholm. San Mateo: Morgan Kaufman.

Gärtner, T. (2003). A survey of kernels for structured data. *SIGKDD Explorations*, *5*(1), 49–58.

Getoor L. & Taskar B. (Eds.) (2007). *Introduction to statistical relational learning*. New York: MIT Press.

Getoor, L., Friedman, N., Koller, D., & Pfeffer, A. (2001a). Learning probabilistic relational models. In S. Džeroski & N. Lavrač (Eds.), *Relational data mining* (pp. 307–335). Dordrecht: Kluwer.

Getoor, L., Segal, E., Taskar, B., & Koller, D. (2001b). Probabilistic models of text and link structure for hypertext classification. In *IJCAI workshop on text learning: beyond supervision*.

Getoor, L., Friedman, N., Koller, D., & Taskar, B. (2002). Learning probabilistic models of link structure. *Journal of Machine Learning Research*, *3*, 679–707.

Heckerman, D., Meek, C., & Koller, D. (2004). *Probabilistic models for relational data* (Technical Report MSR-TR-04-30). Microsoft Research.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (1993). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*, 1527–1554.

Jaeger, M. (1997). Relational Bayesian networks. In M. Kaufmann (Ed.), *Proceedings of the 13'th annual conference on uncertainty in artificial intelligence* (pp. 266–273).

Kersting, K., Raedt, L. D., & Kramer, S. (2000). Interpreting Bayesian logic programs. In *Proceedings of the AAAI-2000 workshop on learning statistical models from relational data* (pp. 29–35), Banff, Alberta, Canada. Menlo Park: AAAI Press.

Kersting, K., Van Otterlo, M., & De Raedt, L. (2004). Bellman goes relational. In *Proceedings of the Twenty-First International Conference on Machine Learning* (pp. 59–67), Banff, Alberta, Canada. Menlo Park: AAAI Press.

King, R., Whelan, K., Jones, F., Reiser, P., Bryant, C., Muggleton, S., Kell, D., & Oliver, S. (2004). Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, *427*, 247–252.

Kok, S., & Domingos, P. (2005). Learning the structure of Markov logic networks. In L. De Raedt & S. Wrobel (Eds.), *Proceedings of the 22'nd annual international conference on machine learning (ICML-2005)* (pp. 441–448). Madison: Omnipress.

Kok, S., & Domingos, P. (2007). Statistical predicate invention. In Z. Ghahramani (Ed.), *Proceedings of the 24'th annual international conference on machine learning (ICML-2007)* (pp. 433–440). Madison: Omnipress.

Koller, D., & Pfeffer, A. (1998). Probabilistic frame-based systems. In *Proceedings of the 14'th annual conference on uncertainty in artificial intelligence* (pp. 580–587).

Kubica, J., Moore, A., Schneider, J., & Yang, Y. (2002). Stochastic link and group detection. In *Proceedings of the 18'th national conference on artificial intelligence* (pp. 798–804). Menlo Park: AAAI Press.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18'th international conference on machine learning* (*ICML-2001*) (pp. 282–289).

Lavrač, N., & Džeroski, S. (1994). *Inductive logic programming: techniques and applications*. Chichester: Ellis-Horwood.

Leathwick, J., Rowe, D., Richardson, J., Elith, J., & Hastie, T. (2005). Using multivariate adaptive regression splines to predict the distributions of New Zealand's freshwater diadromous fish. *Freshwater Biology*, *50*, 2034–2052.

Liang, P., Bouchard-Côté, A., Klein, D., & Taskar, B. (2006). An end-to-end discriminative approach to machine translation. In *Proceedings of the 21'st international conference on computational linguistics* (*COLING/ACL*) (pp. 761–768).

Liben-Nowell, D., & Kleinberg, J. (2003). The link prediction problem for social networks. In *International conference on information and knowledge management* (*CIKM*) (pp. 556–559).

Lowd, D., & Domingos, P. (2005). Naive Bayes models for probability estimation. In L. De Raedt & S. Wrobel (Eds.), *Proceedings of the 22'nd annual international conference on machine learning (ICML-2005)*. New York: Assoc. Comput. Mach.

Lowd, D., & Domingos, P. (2007). Recursive random fields. In *Proceedings of the international joint conference on artificial intelligence* (pp. 950–955). IJCAI.

Lu, Q., & Getoor, L. (2003). Link based classification. In *Proceedings of the 20'th international conference on machine learning*.

Macskassy, S., & Provost, F. (2007). Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning*, *8*, 935–983.

Mark, W., & Perrault, R. (2007). *CALO: a cognitive assistant that learns and organizes* (Technical Report). SRI International.

Mihalkova, L., Huynh, T., & Mooney, R. (2007). Mapping and revising Markov logic networks for transfer learning. In *Proceedings of the 22'nd national conference on artificial intelligence* (pp. 608–614).

Milch, B., & Russell, S. (2006). First-order probabilistic languages: into the unknown. In S. M. R. Otero & A. Tamaddoni-Nezhad (Eds.), *Lecture notes in artificial intelligence: Vol. 4455. Proceedings of the 16th international conference on inductive logic programming* (pp. 10–24). Berlin: Springer.

Milch, B., Marthi, B., & Russell, S. (2004). BLOG: Relational modeling with unknown objects. In *ICML 2004 workshop on statistical relational learning and its connections to other fields*.

Mitchell, T. M., Keller, R. M., & Kedar-Cabelli, S. T. (1986). Explanation-based generalization: A unifying view. *Machine Learning*, *1*(1), 47–80.

Muggleton, S. (1995). Inverse entailment and Progol. *New Generation Computing*, *13*, 245–286.

Muggleton, S. (1996). Stochastic logic programs. In L. de Raedt (Ed.), *Advances in inductive logic programming* (pp. 254–264). Amsterdam: IOS Press.

Muggleton, S. (2005). Machine learning for systems biology. In *LNAI: Vol. 3625. Proceedings of the 15th international conference on inductive logic programming* (pp. 416–423). Berlin: Springer.

Muggleton, S. (2006). Exceeding human limits. *Nature*, *440*(7083), 409–410.

Muggleton, S., & De Raedt, L. (1994). Inductive logic programming: Theory and methods. *Journal of Logic Programming*, *19*, 629–679.

Muggleton, S., & Feng, C. (1990). Efficient induction of logic programs. In *Proceedings of the first conference on algorithmic learning theory* (pp. 368–381). Berlin: Springer.

Neville, J., & Jensen, D. (2000). Iterative classification in relational data. In *AAAI workshop on statistical relational learning*.

Neville, J., & Jensen, D. (2003). Collective classification with relational dependency networks. In *Proceedings of the 2'nd multi-relational data mining workshop*.

Neville, J., & Jensen, D. (2007). Relational dependency networks. *Journal of Machine Learning Research*, *8*, 653–692.

Nocedal, J., & Wright, S. J. (1999). *Numerical optimization*. New York: Springer.

Paes, A., Revoredo, K., Zaverucha, G., & Costa, V. S. (2005). Probabilistic first-order theory revision from examples. In S. Kramer & B. Pfahringer (Eds.), *Lecture notes in artificial intelligence: Vol. 3625. Proceedings of the 15'th international conference on inductive logic programming* (pp. 295–311). Berlin: Springer.

Parker, C., Fern, A., & Tadepalli, P. (2006). Gradient boosting for sequence alignment. In *Proceedings of the 21st national conference on artificial intelligence (AAAI-2006)*, Boston. AAAI Press: Menlo Park.

Parker, C., Fern, A., & Tadepalli, P. (2007). Learning for efficient retrieval of structured data with noisy queries. In Z. Ghahramani (Ed.), *Proceedings of the 24th International Conference on Machine Learning (ICML-2007)* (pp. 729–736). Oregon. Omnipress, Madison: Corvalis.

Pasula, H., Marthi, B., Milch, B., Russell, S., & Shpitser, I. (2002). Identity uncertainty and citation matching. *Advances in Neural Information Processing Systems (NIPS)*, *15*, 1401–1408.

Pfeffer, A. (2001). IBAL: A probabilistic rational programming language. In *Proceedings of the international joint conference on artificial intelligence* (pp. 733–740).

Plotkin, G. (1969). A note on inductive generalisation. In B. Meltzer & D. Michie (Eds.), *Machine intelligence* (Vol. 5, pp. 153–163). Edinburgh: Edinburgh University Press.

Poole, D. (1993). Probabilistic horn abduction and Bayesian networks. *Artificial Intelligence*, *64*(1), 81–129.

Puech, A., & Muggleton, S. (2003). A comparison of stochastic logic programs and Bayesian logic programs. In *IJCAI workshop on learning statistical models from relational data*. IJCAI.

Quinlan, J. (1990). Learning logical definitions from relations. *Machine Learning*, *5*, 239–266.

Reid, M. (2004). Improving rule evaluation using multi-task learning. In R. Camacho, R. King, & A. Srinivasan (Eds.), *Lecture notes in artificial intelligence: Vol. 3194. Proceedings of the 14th international conference on inductive logic programming* (pp. 252–269). Berlin: Springer.

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Machine Learning*, *62*(1–2), 107–136.

Rosenfeld, A., Hummel, R., & Zucker, S. (1976). Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man and Cybernetics*, *SMC-6*, 420–433.

Sanner, S., & Boutilier, C. (2006). Practical linear value-approximation techniques for first-order MDPs. In *Proceedings of the 22'nd annual conference on uncertainty in artificial intelligence*.

Sato, T. (2005). Generative modeling with failure in PRISM. *International joint conference on artificial intelligence* (pp. 847–852). San Mateo: Morgan Kaufmann.

Sato, T., & Kameya, Y. (1997). PRISM: a symbolic-statistical modeling language. In *Proceedings of the 15'th international joint conference on artificial intelligence* (pp. 1330–1335).

Shapiro, E. (1983). *Algorithmic program debugging*. Cambridge: MIT Press.

Sutton, C., & McCallum, A. (2007). Piecewise pseudolikelihood for efficient training of conditional random fields. In Z. Ghahramani (Ed.), *Proceedings of the 24'th international conference on machine learning* (*ICML-2007*) (pp. 863–870). Omnipress.

Tadepalli, P., Givan, B., & Driessens, K. (2004). Relational reinforcement learning: An overview. In *ICML workshop on relational reinforcement learning*, Banff, Canada.

Tamaddoni-Nezhad, A., Chaleil, R., Kakas, A., & Muggleton, S. (2006). Application of abductive ILP to learning metabolic network inhibition from temporal data. *Machine Learning*, *64*, 209–230. doi:10.1007/s10994-006-8988-x.

Tamaddoni-Nezhad, A., Chaleil, R., Kakas, A., Sternberg, M., Nicholson, J., & Muggleton, S. (2007). Modeling the effects of toxins in metabolic networks. *IEEE Engineering in Medicine and Biology*, *26*, 37–46. doi:10.1109/MEMB.2007.335590.

Taskar, B., Segal, E., & Koller, D. (2001). Probabilistic classification and clustering in relational data. In *Proceedings of the international joint conference on artificial intelligence* (pp. 870–878).

Taskar, B., Abbeel, P., & Koller, D. (2002). Discriminative probabilistic models for relational data. In *Proceedings of the 18'th annual conference on uncertainty in artificial intelligence* (pp. 485–492).

Taskar, B., Guestrin, C., & Koller, D. (2003b). Max-margin Markov networks. *Advances in Neural Information Processing Systems*, 16.

Taskar, B., Wong, M., Abbeel, P., & Koller, D. (2003a). Link prediction in relational data. *Advances in Neural Information Processing Systems*, 16.

Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, *6*, 1453–1484.

Wang, C., & Khardon, R. (2007). Policy iteration for relational MDPs. In *Proceedings of the 23'rd annual conference on uncertainty in artificial intelligence*.

Wellman, M., Breese, J., & Goldman, R. (1992). From knowledge bases to decision models. *The Knowledge Engineering Review*, *7*(1), 35–53.

Winston, P. (1975). Learning structural descriptions from examples. In P. Winston (Ed.), *The psychology of computer vision*. New York: McGraw Hill.

Wrobel, S. (1995). First-order theory refinement. In L. D. Raedt (Ed.), *Advances in inductive logic programming* (pp. 14–33). Amsterdam: IOS Press.

Xu, Y., & Fern, A. (2007). On learning linear ranking functions for beam search. In Z. Ghahramani (Ed.) *Proceedings of the 24'th international conference on machine learning* (*ICML-2007*) (pp. 1047–1054). Omnipress.

Xu, Y., Fern, A., & Yoon, S. (2007). Discriminative learning of beam-search heuristics for planning. In M.M. Veloso (Ed.) *Proceedings of the international joint conference on artificial intelligence* (*IJCAI-07*) (pp. 2041–2046). IJCAI.