The Induction and Transfer of Declarative Bias

Will Bridewell
Stanford Center for Biomedical
Informatics Research
Stanford University, Stanford, CA 94305 USA

Ljupčo Todorovski
University of Ljubljana,
Faculty of Administration
Gosarjeva 5, SI-1000 Ljubljana, Slovenia

Abstract

People constantly apply acquired knowledge to new learning tasks, but machines almost never do. Research on transfer learning attempts to address this dissimilarity. Working within this area, we report on a procedure that learns and transfers constraints in the context of inductive process modeling, which we review. After discussing the role of constraints in model induction, we describe the learning method, MISC, and introduce our metrics for assessing the cost and benefit of transferred knowledge. The reported results suggest that cross-domain transfer is beneficial in the scenarios that we investigated, lending further evidence that this strategy is a broadly effective means for increasing the efficiency of learning systems. We conclude by discussing the aspects of inductive process modeling that encourage effective transfer, by reviewing related strategies, and by describing future research plans for constraint induction and transfer learning.

Introduction

Research in transfer learning (Choi *et al*. 2007; Mihalkova *et al*. 2007) promises mechanisms that let induction systems improve with experience even as they operate across different tasks or domains. The wide variety of learning approaches joined with the limited research and lexical ambiguity of this area demands that authors clearly define both the base induction task and the transfer task. In our case, the base task is inductive process modeling (Bridewell *et al*. 2008), which involves learning explanatory models from a mixture of time-series data and domain knowledge. Traditionally, the knowledge comes from human experts and takes the form of model components and structural constraints that limit how those components can be combined (Todorovski *et al*. 2005). The transfer task involves learning new structural constraints during one modeling scenario and using those to reduce search in later problems. This knowledge lets the modeling system rule out candidate structures before calling an expensive parameter estimation routine.

Our approach, implemented as the MISC[1] system, treats constraint induction as an instance of supervised learning. Intuitively, we take advantage of a common property of intelligent systems: they consider several candidate solutions to a problem before returning the most promising ones. By scoring their candidates with an objective function, these systems create a labeled training set that contains implicit information about the search space. From this data, MISC can extract rules that identify the most promising candidates. In our case, an inductive process modeling system (e.g., HIPM) uses these rules, much like its own structural constraints, to limit its search for candidate structures.

Prior applications of this approach have led to constraints that successfully transfer across similar problems (Bridewell & Todorovski 2007a). That is, the transferred constraints can substantially reduce a system's learning costs without a corresponding reduction in model accuracy. However, in those experiments, the problems involved the same biological species living in roughly the same environment. Interestingly, the learned constraints reflected common domain knowledge, a finding that was born out by later work in a more complex scientific domain (Bridewell & Todorovski 2007b). This result in particular suggests that the constraints are general enough for the more difficult task of transferring across domains.[2]

In this paper, we evaluate whether MISC produces transferable, cross-domain knowledge in the context of inductive process modeling. To this end, we consider two aquatic ecosystems—one a lake in a temperate climate and the other a sea in the Southern Ocean. These environments diverge both in their relevant, measured variables and in their environmental influences and are studied by different scientific subfields. We make two conjectures: (1) constraints learned from a source domain will dramatically reduce the number of candidate structures considered in the target domain and (2) these constraints will not substantially reduce solution accuracy compared to search without them. If both hypotheses hold, then we claim that MISC implements a successful approach for cross-domain transfer learning.

The next section provides more details on the base induction task and illustrate the effects of declarative bias in the form of structural constraints. We then introduce MISC and show how it applies to inductive process modeling. Afterwards, we introduce the evaluation measures, report results, and discuss their implications. We close by reviewing related work and suggesting a future line of research.

[1] **M**ethod for **I**nducing **S**tructural **C**onstraints

[2] We recognize that the word "domain" possesses a comforting ambiguity. Within this paper, we discuss domains that differ in their system variables and environmental factors. Notably, we use the same generic process library throughout and do not investigate representation mapping.

Inductive Process Modeling

Scientists use models to explain and predict an observed system's behavior. In the biological sciences, those models commonly refer to processes that govern system dynamics and the entities altered by those processes. Differential equations offer one way to represent these mechanisms and to develop predictive models, but they fail to make contact with the process knowledge shared among scientists. In response, Langley *et al*. (2002) developed a language for *quantitative process models* that ties the explanatory aspects of a model (i.e., the processes and entities) to its mathematical formulation.

This formalism, shown in Table 1, supports both a performance task that requires model simulation and a learning task called inductive process modeling. Simulation involves compiling a process model into a system of differential and algebraic equations and using a numerical solver, such as CVODE (Cohen & Hindmarsh 1996), to produce trajectories for analysis and comparison to observations. The learning task requires domain knowledge in the form of generic processes, generic entities, and constraints on how they may be instantiated and combined. The output is a specific *process model* that explains observed data and predicts unseen data accurately.

The generic processes form the core of the background knowledge. For example,

generic process holling_1:
  variables $S1\{prey\}, S2\{predator\}$;
  parameters $ar[0.01, 10], ef[0.001, 0.8]$;
  equations
    $d[S1.pop, t, 1] = -1 * ar * S1.pop * S2.pop$;
    $d[S2.pop, t, 1] = ef * ar * S1.pop * S2.pop$;

is the generic form of the predation process in Table 1. Notice that we replaced the parameters with continuous ranges and the entities (i.e., f and r) with typed identifiers.

Several systems address this task, including IPM (Bridewell *et al*. 2008) and HIPM (Todorovski *et al*. 2005), and have been shown to learn models with hidden variables, conditionally active processes, and other features. Each system employs a generate-and-test strategy to search for models whose simulated trajectories match the training data. Initially, the generator constructs potential model components by grounding generic processes with specific entities and then carries out a two-layered search through the space of candidate models. The first layer builds an acceptable model structure from available components, and the second estimates the numerical parameters for that structure. The tester simulates each candidate and compares its trajectories to the observations. Combined, the procedures for model construction and evaluation take roughly the same amount of time for each candidate. This approach works well in practice, but has two important drawbacks: parameter estimation accounts for over 99% of the computation per model and model structures may be scientifically implausible. Structural constraints treat both problems.

To illustrate the effects of declarative bias, consider an example problem from predator–prey dynamics. The problem input consists of the population density of interacting rabbit and fox populations over time, rabbit and fox entities that instantiate a generic entity for animal species, and 9 generic processes: 2 for population growth (i.e., exponential and logistic), 2 for population loss, and 5 for predation. Ruling out cannibalism, the generator would produce 18 model components. If we limit each component to a single appearance per model there are $2^{18} - 1 = 262,143$ structures. However, this scenario lets rabbits prey on foxes, which seems implausible. A constraint that rules out fierce bunnies eliminates five of the predation components, leaving $2^{13} - 1 = 8,191$ structures. By further differentiating the species into predator and prey and using common restrictions from population dynamics, the space shrinks to 20 plausible candidates.[3]

Table 1: A process model of a predator–prey interaction between populations of foxes (f) and rabbits (r). The notation $d[X,t,1]$ indicates dX/dt.

model Predation;
entities $r\{prey\}, f\{predator\}$;
process rabbit_growth;
  equations
    $d[r.pop, t, 1] = 1.81 * r.pop * (1 - 0.0003 * r.pop)$;
process fox_death;
  equations
    $d[f.pop, t, 1] = -1 * 1.04 * f.pop$;
process predation_holling_1;
  equations
    $d[r.pop, t, 1] = -1 * 0.03 * r.pop * f.pop$;
    $d[f.pop, t, 1] = 0.30 * 0.03 * r.pop * f.pop$;

This example demonstrates that domain knowledge can dramatically reduce the amount of search carried out by a system. Not only do the constraints exclude several implausible models at the generation stage, but they do so without calling the parameter estimation routine. This advance eliminates a considerable amount of computation. Furthermore, compiling the bias into the generator implicitly bypasses the implausible structures and saves more processor time. However, unlike processes which frequently appear as sets of equations in the scientific literature, modeling constraints tend to be implicit and infrequently codified. As a result, collecting this knowledge requires extensive collaboration with domain experts (Atanasova *et al*. 2006). In the next section, we introduce MISC and show how it can automate the acquisition process in the context of inductive process modeling.

Learning Structural Constraints

The approaches from the previous section fall into an important category of heuristic search techniques. In this class—which includes methods for supervised rule induction, constraint optimization, and equation discovery—each node in the search space constitutes a candidate solution. Although systems that implement this technique typically discard their

[3] A plausible model contains one of two growth processes for rabbits, one of two loss processes for foxes, and one of five for predation. This gives $2 * 2 * 5 = 20$ total models.

search history, adapting them to store visited nodes creates data sets that map potential solutions to their performance. The intuition that informs MISC is that a relationship exists between the form of a solution and its fit to data.

This research rests on two central ideas: (1) one can learn rules that classify structures as accurate or inaccurate for a particular problem and (2) these rules transfer across problems and possibly across domains. We treat the second idea when we report on the experiments. Here, we explore the first idea by treating the task of constraint induction as a supervised learning problem. To this end, MISC takes as input (a) descriptions of models, (b) scores that represents their fit to a data set, and (c) a logical representation of the associated generic process library. The system translates the model descriptions into a base language that indicates which processes and entities appear in each model and how they interact. These descriptions are used by the higher-level relational features shown in Table 2 to characterize the structure of each model.

In addition to describing the models with these features, MISC assigns a target class to each model based on its quantitative fit. To this end, the system selects performance thresholds for the coefficient of determination (r^2), which falls in the [0,1] interval, and labels models that exceed the threshold as *accurate* and those below it as *inaccurate*. To identify candidate thresholds, MISC sorts the models in the training data models by r^2 and divides them into bins. Next, the system identifies the point of maximal performance change between consecutive models within each bin. If points in neighboring bins have similar r^2 scores, the implemented procedure eliminates all but one. We describe how one might select a single threshold from this set when we report on the experiments.

With a feature set and thresholds in hand, MISC relies on inductive logic programming to induce structural constraints. The current implementation uses Aleph (Srinivasan 2000) to generate two separate rule sets for each threshold: one that describes the structure of the accurate models and a corresponding one for the inaccurate models.[4] MISC transforms the rules for accurate structures into a disjunction of necessary conditions for considering a model and transforms those for inaccurate structures into a similar disjunction of conditions for ignoring a model. We can then use these constraints to alter the learning phase of an inductive process modeling system.

Evaluating Constraint Transfer

This paper's hypothesis claims not simply that the constraints learned using MISC help solve future modeling problems within the same domain, but that they successfully transfer to other domains. To test this claim, we need scoring functions to assess the effectiveness of knowledge transfer. Unfortunately, general measures of transfer learning have been elusive since they depend on the nature of the base learning task, the type of knowledge being passed on, and the core objective of transfer. As a result we describe the measures that we use and discuss why we chose them.

[4] We use Aleph's iterative covering algorithm and allow 10 false positives. All other settings take their default values.

Table 2: The domain-general relationships that can appear in a constraint. Each predicate also has an explicit negation.

includes_process(model_id, generic_process)
includes_process_type(model_id, process_type)
includes_entity_instance(model_id, entity_instance)
includes_entity(model_id, generic_entity)
includes_process_entity_instance(model_id, generic_process, entity_instance)
includes_process_entity(model_id, generic_process, generic_entity)
includes_process_type_entity_instance(model_id, process_type, entity_instance)
includes_process_type_entity(model_id, process_type, generic_entity)

We can restate our conjectures as two questions:

(1) How much search time do you save?

(2) How much accuracy do you lose?

The first of these addresses the tradeoff between declarative bias and computation. Intuitively, more knowledge should equal less search. Since inductive process modeling uses constrained, exhaustive search, we can directly measure this effect. The second question addresses the primary danger of adding bias. That is, the constraints could rule out the best solutions. Since the base task carries out exhaustive search, and since the constraints reduce the search space to a proper subset of candidate structures, the results cannot improve. The answers to these questions tell us (1) the transfer benefit and (2) the transfer cost.

Prior work in this area (Bridewell & Todorovski 2007a) reported the transfer benefit by directly comparing the number of models in the search space with and without the transferred knowledge. For example, a score of 9.0 indicated that the additional bias led to a 9-fold reduction in search. We modify that measure to produce a value that falls between zero, which indicates no reduction, and one, which indicates an empty search space. Specifically, we report the fraction of the original search space that the structural constraints rule out: $1 - \frac{t_n}{nt_n}$, where t_n is the number of models in the transfer scenario and nt_n is the number in the nontransfer scenario.

Our measure of transfer cost also diverges from earlier work. Since we are interested in a single solution, measuring the recall of the top n models (Bridewell & Todorovski 2007a) can lead to overly pessimistic results for two reasons. First, if the bias rules out all but the highest scoring model, we should not penalize it for failing to retrieve the top 10, 50, or 100. Second, in practice, the highest score and the 50th highest score may differ by 1/100th of a point which may be negligible compared to uncertainty in the data. To avoid these problems, we calculate transfer cost as $(1 - \frac{t_{best}}{nt_{best}})$ where t_{best} is the r^2 score for the highest ranked model in the transfer condition and nt_{best} is the highest ranked model in the unconstrained search space. A cost

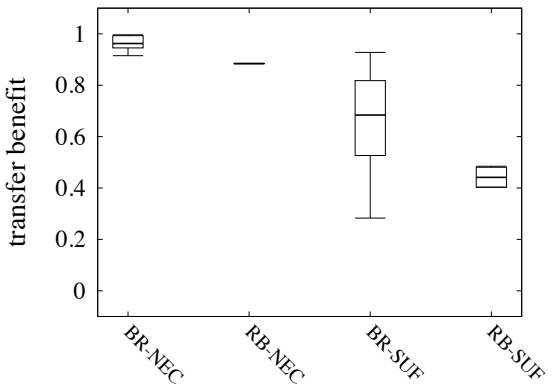

Figure 1: The *transfer benefit* for applying necessary (NEC) and sufficient (SUF) constraints from Bled Lake to the Ross Sea (BR) or vice versa (RB). The boxes identify the 1st, 2nd, and 3rd quartiles, and the lines delineate the minimum and maximum values.

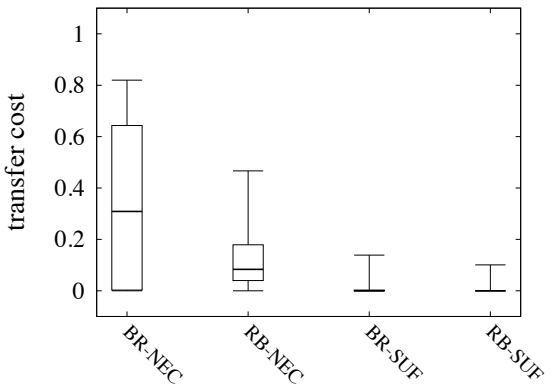

Figure 2: The *transfer cost* for applying necessary (NEC) and sufficient (SUF) constraints from Bled Lake to the Ross Sea (BR) or vice versa (RB). The boxes identify the 1st, 2nd, and 3rd quartiles, and the lines delineate the minimum and maximum values.

of 0 indicates that no accuracy was lost, whereas a cost of 1 corresponds to the extreme case where the constraints left only those models with an r^2 of 0. Coupled with our measure of transfer benefit, this measure of transfer cost lets us evaluate MISC in the context of ecological modeling.

Results from Cross-Domain Transfer

To evaluate the practical effect of constraint learning on inductive process modeling, we applied it to the task of explaining ecosystem dynamics. Models of population dynamics identify the mechanisms that drive observed changes in species concentrations within an ecosystem. These mechanisms may include processes of nutrient uptake for plants, grazing for herbivores, and predation for carnivores as possible examples. For this study, we focused our attention on the general domain of aquatic ecosystems, which includes both freshwater and marine (i.e., saltwater) environments. Although these ecosystem types share more in common with each other than with terrestrial systems, they differ substantially in their species composition and in the presence and the dynamics of the relevant environmental factors.

The two ecosystems that we selected differ not only along the freshwater–saltwater dimension but also in geographical location. The marine data are from the 1996–1997 and 1997–1998 summer seasons of the Ross Sea (Arrigo *et al*. 2008), which lies in the Southern Ocean off the coast of Antarctica. In this scenario, we had observations of five variables: phytoplankton concentration, nitrate concentration, ice concentration, available light, and water temperature. To these we added unmeasured variables for the zooplankton and iron concentrations—two properties thought to affect the phytoplankton bloom. In comparison, the freshwater data came from the Bled Lake (Atanasova *et al*. 2006) in Slovenia from 1995–2002 during the months where the lake was not covered in ice. Observations from this site covered seven variables: the amount of available light, the water temperature, and the concentrations of phytoplankton, zooplankton, nitrate, phosphate, and silica. These regions differ in the particular plankton species, the set of operative nutrients, and the expressed environmental factors (e.g., there is no night during the summer in the Ross Sea). Nevertheless, we expected the same general processes to drive the underlying dynamics and that this commonality would support constraints that transfer across their parent scientific domains. Each year or season serves as a data set, which gives us two sets for the Ross Sea and eight for Bled Lake.

For the experimental evaluation, we used HIPM (Todorovski *et al*. 2005) to exhaustively explore the space of models that could explain the Ross Sea and Bled Lake data sets and MISC to induce transferable knowledge. In each case we employed the same domain knowledge for aquatic ecosystems which included 29 generic processes, 6 generic entities, and structural constraints. This library was assembled and validated by an ecologist in collaboration with scientists who study the Bled Lake and the Ross Sea. The instantiated entities differed between the Ross Sea and Bled Lake and affected the number of candidate solutions for each domain: 1108 for the Ross Sea and 3120 for Bled Lake. After running HIPM on all 10 data sets individually, we applied MISC at each of its suggested thresholds. This produced two sets of rules for each run: one that described accurate models, which became *necessary* constraints, and another that described the inaccurate models, which were negated to gain *sufficient* constraints.

To transfer the knowledge across domains, we implemented a selection procedure that chose constraint sets based on within-domain problems and evaluated them on the cross-domain problems. For example, we took the set of necessary constraints derived from the 1995 Bled Lake data for each threshold and applied them to the remaining Bled Lake years. The constraint set that generalized best to the other years was selected for transfer. Our selection criteria involved picking the set of constraints that maximized $(benefit + 1 - cost)/2$ for other problems within the same domain. This step led to separate collections of necessary constraints and sufficient constraints for each of 10 modeling

problems. Our assumption was that a set of constraints that works well within a domain would also work well across domains. Following this approach, there were 2 modeling tasks for the Ross Sea domain and 8 for the Bled Lake, which gave 16 cases of transfer in each direction. Since we considered the necessary and sufficient constraints separately, we had a total of 64 transfer experiments.

The following rules are an example of MISC's output in a Prolog format. The first rule states that if phytoplankton is modeled with a second-order exponential death process and without the specified grazing process then the candidate structure is inaccurate. The second rule states that if a model includes a limited growth process and the generalized Gause grazing process, then it is accurate.

inaccurate model(M) :-
  includes_process_entity(M, death_exp2, phy),
  does_not_include_process_entity(M, holling_type_3, phy).
accurate model(M) :-
  includes_process_type(M, limited_growth),
  includes_process(M, generalized_gause).

These rules provide the constraints that reduce the search space of an inductive process modeling system.

Figure 1 summarizes the transfer benefit and Figure 2 summarizes the transfer cost. In 68.8% of the cases the benefit exceeded 0.5, which means that the added bias cut the search space by 50% or more. Also, 31.3% of the cases had a benefit greater than 0.9, which is an order of magnitude reduction. The average benefit was 0.92 when using the necessary constraints and 0.55 when applying the sufficient ones. Overall, the cost was less than 0.1 for 70.3% of the time and 0.0 for 40.6% of the time. The average cost when using the necessary constraints was 0.25 and 0.01 when applying the sufficient ones.

Discussion

The results with transfer benefit imply a substantial reduction in computation due to the learned constraints. The current results are, in some cases, more modest than the order-of-magnitude reductions shown in previous studies (Bridewell & Todorovski 2007a). We attribute that finding to the fact that the aquatic ecosystem library already contains extensive, expert-derived constraints that make the modeling task manageable. As a result, we were surprised to find further constraints that complemented the expert knowledge and generalized across domains. Based on that prior study, we conjecture that domains lacking a strong bias on the structure of solutions will see higher transfer benefit.

Although there were some exceptions, transfer cost was low. This result means that the added bias either preserved the highest scoring model (41% of the time) or a model with a similar score (70% of the time). Notably, transfer cost was minimal for the sufficient constraints, but could be quite high for the necessary ones. The cause of this discrepancy could be related to the technologies in MISC, or it could result from properties of the modeled domains. To uncover the source of this effect and its ubiquity, we will need to examine data from a broad range of transfer scenarios. However, the parameters in Aleph that control rule selection (e.g., precision and coverage) may also be dominant factors.

Finally, we point out the differences in the results that relate to the direction of transfer. In Figures 1 and 2, transfer from Bled Lake generally reflects a wider distribution of costs and benefits than transfer from the Ross Sea. This relates to the number of constraint sets in each case. There were 16 total constraint sets derived from the Bled Lake task and only 4 from the Ross Sea one. Constraints learned from a single modeling task in one domain (e.g., Bled Lake, 1995) will remove the same number of candidate structures from the other domain regardless of the year being modeled. As the number of source domains increases and the number of target domains decreases the variability in the constraint sets has a greater effect on the variance. We posit that a similar result explains the differences in transfer cost.

The findings in this paper build upon and strengthen prior work by Bridewell & Todorovski (2007a) in three ways. First, our experiments evaluated the potential for cross-domain transfer as opposed to within-domain transfer. Second, our modeling problems were more complex than basic predator–prey interactions, involving unobserved variables and a wider variety of potential interactions. Third, they constrained their models by the number of processes, whereas our ecosystem library included several structural constraints that were fine-tuned by a domain expert. As a result, there is strong evidence for our two conjectures, that MISC dramatically reduces the number of structures considered in the target domain and that solution accuracy will remain stable.

Related Work and Concluding Remarks

Notably, without a rich representation transfer of any sort is difficult. Work investigating multitask learning (e.g., Zhang *et al*. 2008) typically uses a propositional language for input and lacks background knowledge about task structure. As a result, much of that research is directed toward representations that enable the induction of the shared structure. Caruana (1997) rightly suggests that we need a theory of task relatedness to better understand multitask and transfer learning, but any such theory that fails to take the task structure as input will be of limited use.

Our research more closely resembles methods for transferring and adapting relational knowledge across domains and for inducing structures with general applicability. For example, TAMAR (Mihalkova *et al*. 2007) maps first-order logical predicates across domains, transfers the corresponding theories, and then revises the structure to apply to the new scenario. Whereas that system learns by transferring and then adapting a solution from another domain, MISC attempts to induce knowledge about solutions that is suitable for transfer. DTM (Davis & Domingos 2009) goes one step further by identifying domain independent patterns that may be useful in many scenarios.

MISC also shares similarities with research outside the area of transfer learning. Bessiere *et al*. (2006) present a system that learns constraint networks using a version space approach, but these networks are propositional and the emphasis is on automated programming as opposed to con-

straint transfer. In addition, systems that learn search control knowledge, such as PRODIGY (Carbonell *et al*. 1991) and SAGE.2 (Langley 1983), identify constraints that reduce the size of the search space. However, their constraints tend to be domain and problem specific and evaluate local decisions in a problem-solving context as opposed to candidate solutions during an exhaustive search.

We see a wide vista of research opportunities based on MISC. One item on our agenda involves incorporating a representation mapping component so that we can improve our support for scenarios where the entities differ. Research on this front could also let us explore situations where the generic processes may take different forms. In that case, we can infer similarities in processes based on the equations that define their behavior and other key characteristics. We also plan to study constraint induction in scenarios where exhaustive search is impossible and to investigate alternative approaches to constraint induction.

More broadly, we believe that the techniques used by MISC apply to a variety of problems in artificial intelligence (AI) including rule induction and constraint optimization. Adapting this approach requires the development of a language that describes the structure of candidate solutions. With this in hand, the basic strategy of using inductive logic programming to infer declarative bias should easily transfer. Much of the labor would involve either casting the AI problem into a constraint-satisfaction mold to take advantage of established technology (Selman, Levesque, & Mitchell 1992) or compiling the constraints into an existing solution generator. In the future, we intend to expand MISC to address some of these other areas.

Inductive process modeling uses rich background knowledge that generalizes across domains, and we have described a method for automatically extending that knowledge. We evaluated a system that implements this method using measures that capture the cost and benefit of transfer, and the experiments showed that the learned constraints dramatically reduced search with minimal effects on solution accuracy.

Acknowledgments

This research was supported by National Science Foundation Grant No. IIS-0326059. We thank Pat Langley and Tolga Könik for helpful discussions; and Kevin Arrigo, Gert van Dijken, and Stuart Borrett for data and domain expertise.

References

Arrigo, K.; van Dijken, G. L.; and Bushinsky, S. 2008. Primary production in the Southern Ocean, 1997–2006. *Journal of Geophysical Research* 113:C08004.

Atanasova, N.; Todorovski, L.; Džeroski, S.; and Kompare, B. 2006. Constructing a library of domain knowledge for automated modelling of aquatic ecosystems. *Ecological Modelling* 194:14–36.

Bessiere, C.; Coletta, R.; Koriche, F.; and O'Sullivan, B. 2006. Acquiring constraint networks using a SAT-based version space algorithm. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, 23–24. Boston, MA: AAAI Press.

Bridewell, W., and Todorovski, L. 2007a. Learning declarative bias. In *Proceedings of the Seventeenth Annual International Conference on Inductive Logic Programming*, 63–77. Corvallis, OR: Springer.

Bridewell, W., and Todorovski, L. 2007b. Extracting constraints for process modeling. In *Proceedings of the Fourth International Conference on Knowledge Capture*, 87–94. Whistler, BC: ACM Press.

Bridewell, W.; Langley, P.; Todorovski, L.; and Džeroski, S. 2008. Inductive process modeling. *Machine Learning* 71:1–32.

Carbonell, J.; Etzioni, O.; Gil, Y.; Joseph, R.; Knoblock, C.; Minton, S.; and Veloso, M. 1991. PRODIGY: an integrated architecture for planning and learning. *ACM SIGART Bulletin*, 2:51–55.

Caruana, R. 1997. Multitask learning. *Machine Learning* 28:41–75.

Choi, D.; Könik, T.; Nejati, N.; Park, C.; and Langley, P. 2007. Structural transfer of cognitive skills. In *Proceedings of the Eighth International Conference on Cognitive Modeling*, 115–120. Ann Arbor, MI: Taylor & Francis Group.

Cohen, S., and Hindmarsh, A. 1996. CVODE, a stiff/nonstiff ODE solver in C. *Computers in Physics*, 10:138–143.

Davis, J., and Domingos, P. 2009. Deep transfer via second-order Markov logic. In *Proceedings of the Twenty-Sixth International Conference on Machine Learning*, 217–224. Montreal, Canada.

Langley, P. 1983. Learning effective search heuristics. In *Proceedings of the Eighth international Joint Conference on Artificial intelligence*, 419–421, Karlsruhe, Germany: Morgan Kaufmann.

Langley, P.; Sánchez, J.; Todorovski, L.; and Džeroski, S. 2002. Inducing process models from continuous data. In *Proceedings of the Nineteenth International Conference on Machine Learning*, 347–354, Sydney, Australia: Morgan Kaufmann.

Mihalkova, L.; Huynh, T.; and Mooney R. J. 2007. Mapping and revising Markov logic networks for transfer learning. In *Proceedings of the Twenty-Second National Conference on Artificial Intelligence*, 608–614. Vancouver, Canada: AAAI Press.

Selman, B.; Levesque, H.; and Mitchell, D. 1992. A new method for solving hard satisfiability problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 440–446. San Jose, CA: AAAI Press.

Srinivasan, A. 2000. *The Aleph Manual*. Computing Laboratory, Oxford University.

Todorovski, L.; Bridewell, W.; Shiran, O.; and Langley, P. 2005. Inducing hierarchical process models in dynamic domains. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, 892–897. Pittsburgh, PA: AAAI Press.

Zhang, J.; Ghahramani, Z.; and Yang, Y. 2008. Flexible latent variable models for multi-task learning. *Machine Learning* 73:221–242.