



Deep Transfer as Structure Learning in Markov Logic Networks

David Moore and Andrea Danyluk

{10dam, andrea}@cs.williams.edu

Williams College

Williamstown, MA

Introduction

Markov logic networks (MLNs) generalize first-order logic and probabilistic graphical models, using weighted formulas of first-order logic to represent relational knowledge. The deep transfer algorithm (DTM) proposed by Davis and Domingos (2009) claims to improve structure learning in MLNs by identifying domain-independent abstract knowledge, represented by second-order logical cliques which capture structural regularities of first-order clauses modeling the source domain, and can be used to bias the search for clauses which model the target domain. Compared to an existing structure learning method, DTM has been shown to deliver improved performance for several pairings of source and target domains.

We exhibit CSGL, a modification of DTM which achieves performance improvements equaling or surpassing those obtained by DTM, but without making use of any external knowledge. This suggests that performance increases observed through DTM are not due primarily to its role in transferring knowledge across domains. Instead, the clique-scoring step of the algorithm can be seen as part of a novel, standalone method for structure learning.

Data and Methods

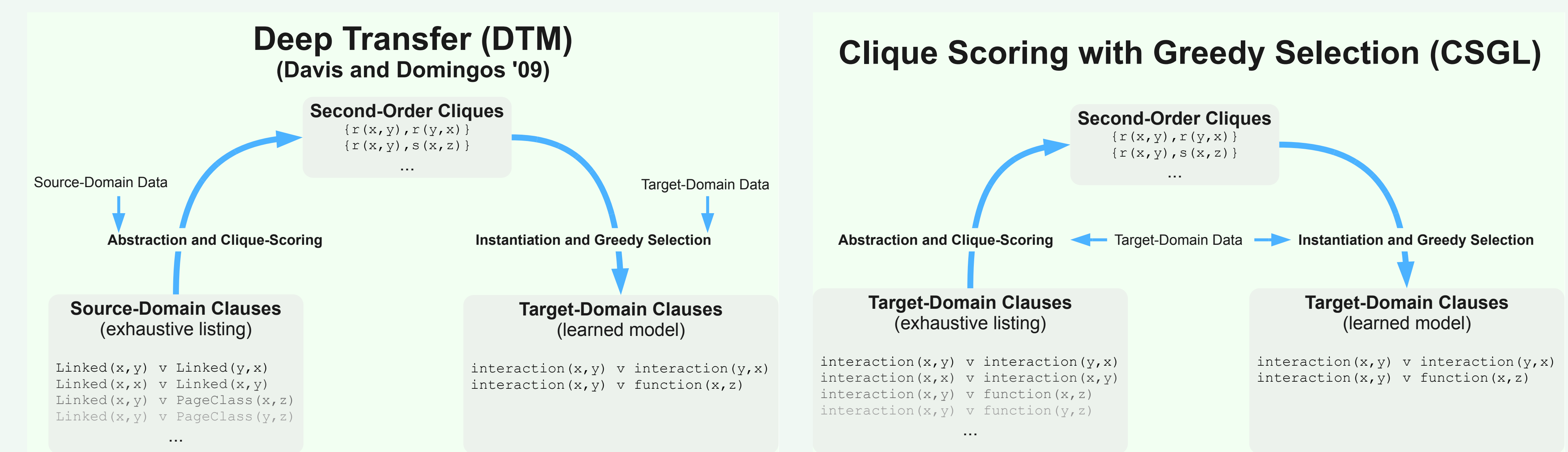
Domains:

- **IMDB:** predicates indicate the relationships between movies, actors, directors, etc. from the Internet Movie Database (imdb.com).
- **UW-CSE:** anonymized relationships between students, faculty, and courses in the University of Washington Computer Science and Engineering Department.
- **WebKB:** labeled web pages from the computer science departments of four universities. Predicates indicate the words occurring on each page, the class label of each page (faculty, student, course, etc.), and the links between pages.
- **Yeast Protein:** protein locations, functions, phenotypes, classes, and enzymes within the yeast *Saccharomyces cerevisiae*, as well as protein interactions and protein complex data, from the MIPS Comprehensive Yeast Genome Database.

Each dataset is divided into four independent folds; we performed cross-validation by training on every subset of three folds and testing on the fourth. The task is to predict the truth values of all ground atoms in the test set which contain particular predicates, using as evidence for each atom the truth values of all other ground atoms in the test set.

Source clauses for DTM were generated by exhaustive search over all clauses containing at most three literals and three object variables. Results are reported using area under the precision-recall curve (AUC). In the results, the “MSL” column denotes baseline structure learning using the beam-search-based algorithm of Kok and Domingos (2005).

The CSGL Algorithm: Structure Learning through Self-Transfer



The CSGL (Clique Scoring with Greedy Selection) algorithm, shown at right and described below, modifies deep transfer to perform *self-transfer*, i.e. it uses the same domain as both source and target. CSGL proceeds by the following steps:

- 1) First, list all first-order clauses in the target domain, up to some maximum length.
- 2) Abstract these clauses into second-order cliques and score them via the DTM clique-scoring process.
- 3) Instantiate the k top-scoring cliques as first-order clauses of the target domain.
- 4) Use the “greedy transfer without refinement” method described by Davis and Domingos (2009) to derive a final set of clauses indicating the structure of the domain.

The absence of the refinement step differentiates CSGL from the DTM self-transfer reported in the first table below, and establishes CSGL as a novel standalone structure learning algorithm.

Experimental Results

This table gives results for DTM with refinement ($k = 10$), performing transfer to each target predicate from each of the four source domains. Cases of self-transfer are in **blue**, and the best result(s) for each predicate are **bolded**:

| | IMDB | UW-CSE | WebKB | Yeast | MSL |
|-----------------------------|-------------|-------------|-------------|-------------|-------------|
| WorkedInGenre (IMDB) | 0.63 | 0.61 | 0.61 | 0.61 | 0.32 |
| WorkedUnder (IMDB) | 0.77 | 0.23 | 0.21 | 0.21 | 0.03 |
| AdvisedBy (UW-CSE) | 0.08 | 0.08 | 0.08 | 0.08 | 0.04 |
| Linked (WebKB) | 0.01 | 0.01 | 0.09 | 0.09 | 0.004 |
| PageClass (WebKB) | 0.86 | 0.86 | 0.68 | 0.68 | 0.87 |
| Function (Yeast) | 0.34 | 0.34 | 0.33 | 0.33 | 0.27 |
| Interaction (Yeast) | 0.04 | 0.04 | 0.10 | 0.10 | 0.04 |

For no predicate does the best case of cross-domain transfer perform significantly better than self-transfer (paired one-tail t -test, $p > 0.10$). This is consistent with our claim that the gains of DTM over MSL are not related to DTM’s use of source-domain knowledge.

Next, we compare the CSGL algorithm (with $k = 5$ and $k = 10$) directly to the baseline structure learner:

| | CSGL-5 | CSGL-10 | MSL |
|-----------------------------|-------------|-------------|-------------|
| WorkedInGenre (IMDB) | 0.70 | 0.63 | 0.32 |
| WorkedUnder (IMDB) | 0.26 | 0.69 | 0.03 |
| AdvisedBy (UW-CSE) | 0.04 | 0.06 | 0.04 |
| Linked (WebKB) | 0.06 | 0.06 | 0.004 |
| PageClass (WebKB) | 0.86 | 0.86 | 0.87 |
| Function (Yeast) | 0.31 | 0.31 | 0.27 |
| Interaction (Yeast) | 0.10 | 0.10 | 0.04 |

CSGL-10 beats MSL in every case except for the PageClass predicate of WebKB, for which the two methods give approximately equal results. This shows that the costly refinement step used above is not necessary to achieve good results, and therefore that CSGL performs well as a standalone structure learner.

Related Work

As a structure-learning algorithm, CSGL bears strong similarities to LHL-FindPaths, the unlifted variant of the hypergraph-lifting approach described by Kok and Domingos (2009). Indeed, the methods proceed in parallel steps:

- 1) The exhaustive search process used by CSGL to generate initial clauses is equivalent to the process in LHL-FindPaths of enumerating and variabilizing paths in the unlifted hypergraph, except for the added restriction that every conjunction which LHL-FindPaths considers must have at least one support in the data.
- 2) Both methods evaluate clauses according to how well they represent structural regularities not found in their sub-clauses; in CSGL this is implemented by the clique-scoring process in which all but the top-scoring cliques are discarded, while in LHL this is done by simply discarding any clause having a smaller weighted-pseudo-log-likelihood (WPLL) than one of its subclauses.
- 3) Both methods consider as candidates many combinations of negated and non-negated atoms in the clauses that they generate; in CSGL this is part of the clique abstraction and instantiation process, while LHL explicitly constructs partially-negated variants of its clauses.
- 4) Both methods arrive at the final MLN structure by greedily selecting clauses from a list of candidates until no clause further improves the overall WPLL.

These parallels provide strong intuition for understanding CSGL, and therefore DTM, as a structure learning algorithm. They also imply that we should not expect that CSGL will provide a significant advance in the state of the art for structure learning. However, it does provide a reference point in design space which might help to guide the development of future structure learning algorithms.

Future Work

Possible directions for future research include analysis of circumstances in which DTM might still benefit from cross-domain transfer (e.g. when the target domain has very little data), identifying other transfer learning mechanisms for which a similar self-transfer trick could be applied to improve single-task learning performance, and exploring whether clique scoring might be integrated with LHL as a more sophisticated approach to identifying useful clauses.

References

- Davis, J., and Domingos, P. 2009. Deep transfer via second-order Markov logic. In *Proceedings of ICML-2009*.
- Kok, S., and Domingos, P. 2005. Learning the structure of Markov logic networks. In *Proceedings of ICML-2005*, 441–448.
- Kok, S., and Domingos, P. 2009. Learning Markov logic network structure via hypergraph lifting. In *Proceedings of ICML-2009*.

Acknowledgments

Special thanks to Jesse Davis for providing us with his implementation of DTM, the WebKB and Yeast datasets, and a great deal of helpful advice.