Data-Driven Theory Refinement Algorithms for Bioinformatics

Jihoon Yang
Information Sciences Lab
HRL Laboratories
3011 Malibu Canyon Road
Malibu, CA 90265
yang@wins.hrl.com

Rajesh Parekh
Data Mining Group
Allstate Research & Planning Center
321 Middlefield Road
Menlo Park, CA 94025
rpare@allstate.com

Vasant Honavar
AI Lab & Computational Biology Lab
Computer Science Department
Iowa State University
Ames, IA 50011
honavar@cs.iastate.edu

Drena Dobbs
Computational Biology Lab
Zoology & Genetics Department
Iowa State University
Ames, IA 50011
ddobbs@iastate.edu

Abstract— **Bioinformatics and related applications call for efficient algorithms for knowledge-intensive learning and data-driven knowledge refinement. Knowledge based artificial neural networks offer an attractive approach to extending or modifying incomplete knowledge bases or domain theories. We present results of experiments with several such algorithms for data-driven knowledge discovery and theory refinement in some simple bioinformatics applications. Results of experiments on the ribosome binding site and promoter site identification problems indicate that the performance of KBDistAl and Tiling-Pyramid algorithms compares quite favorably with those of substantially more computationally demanding techniques.**

I. Introduction

Inductive learning systems attempt to learn a concept description from a sequence of labeled examples [1], [2]. Artificial neural networks, because of their massive parallelism and potential for fault and noise tolerance, offer an attractive approach to inductive learning [1], [3], [4]. Such systems have been successfully used for data-driven knowledge acquisition in several application domains. However, these systems generalize from the labeled examples alone. The availability of domain specific knowledge (domain theories) about the concept being learned can potentially enhance the performance of the inductive learning system [5]. Hybrid learning systems that effectively combine domain knowledge with the inductive learning can potentially learn faster and generalize better than those based on purely inductive learning (learning from labeled examples alone). In practice the domain theory is often *incomplete* or even *inaccurate*.

Inductive learning systems that use information from training examples to modify an existing domain theory by either augmenting it with new knowledge or by refining the existing knowledge are called *theory refinement* systems.

Theory refinement systems can be broadly classified into the following categories.

- **Approaches based on Rule Induction** which use decision tree or rule learning algorithms for theory revision. Examples of such systems include RTLS [6], EITHER [7], PTR [8], and TGCI [9].
- **Approaches based on Inductive Logic Programming** which represent knowledge using first-order logic (or restricted subsets of it). Examples of such systems include FOCL [10] and FORTE [11].
- **Connectionist Approaches using Artificial Neural Networks** which typically operate by first embedding domain knowledge into an appropriate initial neural network topology and refine it by training the resulting neural network on the set of labeled examples. The KBANN system [12], [13] as well as related approaches [14] and [15] offer examples of this approach.

In experiments involving datasets from the Human Genome Project[1], KBANN has been reported to have outperformed symbolic theory refinement systems (such as EITHER) and other learning algorithms such as backpropagation and ID3 [13]. KBANN is limited by the fact that it does not modify the network's topology and theory refinement is conducted solely by updating the connection weights. This prevents the incorporation of new rules and also restricts the algorithm's ability to compensate for inaccuracies in the domain theory. Against this background, constructive neural network learning algorithms, because of their ability to modify the network architecture by dynamically adding neurons in a controlled fashion [16], [17], [18], offer an attractive approach to data-driven theory re-

This research was partially supported by grants from the National Science Foundation (IRI-9409580) and the John Deere Foundation to Vasant Honavar and a grant from the Carver Foundation to Drena Dobbs and Vasant Honavar.

[1]These datasets are available at ftp://ftp.cs.wisc.edu/machine-learning/shavlik-group/datasets/.

0-7803-5529-6/99/$10.00 ©1999 IEEE

finement. Available domain knowledge is incorporated into an initial network topology (e.g., using the rules-to-network algorithm of [12] or by other means). Inaccuracies in the domain theory are compensated for by extending the network topology using training examples. Figure 1 depicts this process.

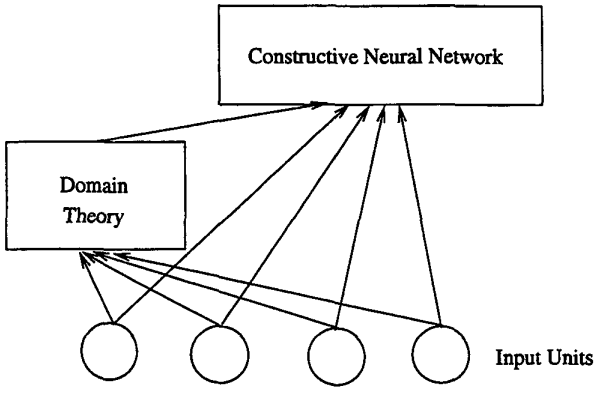

Fig. 1. Theory Refinement using a Constructive Neural Network

II. Constructive Theory Refinement Using Knowledge-Based Neural Networks

This section briefly describes the constructive theory refinement systems which are experimentally compared in this paper on some sample data-driven knowledge refinement tasks in bioinformatics.

Fletcher and Obradović [19] designed a constructive learning method for dynamically adding neurons to the initial knowledge based network. Their approach starts with an initial network representing the domain theory and modifies this theory by constructing a single hidden layer of threshold logic units (TLUs) from the labeled training data using the HDE algorithm [20]. The HDE algorithm divides the feature space with hyperplanes. Fletcher and Obradović's algorithm maps these hyperplanes to a set of TLUs and then trains the output neuron using the pocket algorithm [21].

The RAPTURE system is designed to refine domain theories that contains probabilistic rules represented in the certainty-factor format [22]. RAPTURE's approach to modifying the network topology differs from that used in KBDistAl as follows: RAPTURE uses an iterative algorithm to train the weights and employs the information gain heuristic [23] to add links to the network. KBDistAl is simpler than RAPTURE in that it uses a non-iterative constructive learning algorithm to augment the initial domain theory.

Opitz and Shavlik have extensively studied connectionist theory refinement systems that overcome the fixed topology limitation of the KBANN algorithm [24], [25]. The TopGen algorithm [24] uses a heuristic search through the space of possible expansions of a KBANN network constructed from the initial domain theory. TopGen maintains a queue of candidate networks ordered by their test accuracy on a cross-validation set. At each step, TopGen picks the best network and explores possible ways of expanding it. New networks are generated by strategically adding nodes at different locations within the best network selected. These networks are trained and inserted into the queue and the process is repeated.

The REGENT algorithm uses a genetic search to explore the space of network architectures [25]. It first creates a diverse initial population of networks from the KBANN network constructed from the domain theory. Genetic search uses the classification accuracy on a cross-validation set as a fitness measure. REGENT's mutation operator adds a node to the network using the TopGen algorithm. It also uses a specially designed crossover operator that maintains the network's rule structure. The population of networks is subjected to fitness proportionate selection, mutation, and crossover for many generations and the best network produced during the entire run is reported as the solution. The KBDistAl algorithm [26], constructs a single hidden layer. It uses a computationally efficient DistAl algorithm which constructs the entire network in one pass through the training set instead of relying on the iterative approach used by Fletcher and Obradovć which requires a large number of passes through the training set. The key idea behind DistAl [17], [18], [27] is to add *hyperspherical* hidden neurons one at a time based on a greedy strategy which ensures that each hidden neuron that is added correctly classifies a maximal subset of training patterns belonging to a single class. Correctly classified examples can then be eliminated from further consideration. The process is repeated until the network correctly classifies the entire training set (or some other suitable termination criterion e.g., based on cross-validation is met). It is straightforward to show that DistAl which sets the hidden to output layer weights without going through an iterative process is guaranteed to converge to 100% classification accuracy on any finite training set in time that is quadratic in the number of training patterns [18]. Experiments reported in [18] show that DistAl, despite its simplicity, yields classifiers that compare quite favorably with those generated using more sophisticated (and substantially more computationally demanding) learning algorithms. KBDistAl uses a very simple approach to incorporating prior knowledge into DistAl. The input patterns are classified using the rules and the resulting outputs are used to augment the pattern before it is fed to the neural network which is constructed using DistAl. That is, DistAl is used as the constructive algorithm in Figure 1. This eliminates the need for translating the rules into a neural network.

The Tiling-Pyramid algorithm [28] uses a novel combination of the Tiling and Pyramid constructive learning algorithms [16], [21]. It employs a symbolic knowledge encoding procedure to translate a domain theory into a set of propositional rules using a procedure that is based on the *rules-to-networks* algorithm of Towell and Shavlik [12] which is used in KBANN, TopGen, and REGENT. It yields a set of rules each of which has only one antecedent. The rule set is then mapped to an AND-OR graph which in turn is directly translated into a neural network. The Tiling-Pyramid algorithm uses the Tiling algorithm to construct a *faithful*

representation of the pattern set. A faithful representation of the pattern set is one in which no two patterns belonging to different output classes has the same output representation. The Pyramid algorithm is used to successively add new layers of TLUs to the network. Each newly added layer becomes the network's new output layer. The neurons of the new layer are connected to the input layer and all previously added layers of the network. Experiments have shown that the hybrid algorithm outperforms both Pyramid and Tiling algorithms [16].

In the experiments reported in this paper, the Tiling-Pyramid constructive learning algorithm was used to augment the initial domain knowledge. The hybrid network was trained using the *Thermal Perceptron* learning rule [29]. Each TLU was trained for 1000 epochs with the initial weights chosen randomly between -1 and 1 and the initial temperature for the thermal perceptron (T_0) set to 10.0.

III. Experimental Results

This section reports results of experiments using KBDistAI and Tiling-Pyramid on data-driven theory refinement on the ribosome binding site and promoter site prediction used by Shavlik's group [5], [12], [13], [24], [25]:

- **Ribosome**
  This data is from the Human Genome Project. It comprises of a domain theory and a set of labeled examples. The input is a short segment of DNA nucleotides, and the goal is to learn to predict whether the DNA segments contain a ribosome binding site. There are 17 rules in the domain theory, and 1880 examples in the dataset.
- **Promoters**
  This data is also from the Human Genome Project, and consists of a domain theory and a set of labeled examples. The input is a short segment of DNA nucleotides, and the goal is to learn to predict whether the DNA segments contain a promoter site. There are 31 rules in the domain theory, and 940 examples in the dataset.

The reported results are based on a 10-fold cross-validation. The average training and test accuracies of the rules in domain theory alone were 87.29 ± 0.22 and 87.29 ± 2.03 for **Ribosome** dataset and 77.45 ± 0.56 and 77.45 ± 5.01 for **Promoters** dataset, respectively. Table I and II shows the average generalization accuracy and the average network size (along with the standard deviations where available) for **Ribosome** and **Promoters** datasets, respectively.

Table I and II compare the performance of KBDistAI and Tiling-Pyramid with that of some of the other approaches that have been reported in the literature. Tiling-Pyramid substantially outperforms TopGen, REGENT and KBDistAI in terms of generalization accuracy on the Promoters dataset. Tiling-Pyramid's generalization accuracy is comparable to that of TopGen and REGENT and substantially better than that of KBDistAI on the Ribosome dataset. We conjecture that KBDistAI might have suffered from overfitting of training data in the case of the Ribosome data. In fact, when the network pruning procedure was applied in KBDistAI to avoid overfitting, the generalization accuracy became comparable to that of Tiling-Pyramid.

TABLE I
Results of **Ribosome** dataset.

	Test %	*Size*
Rules alone	87.3 ± 2.0	–
KBDistAI (no pruning)	86.3 ± 2.4	40.3 ± 1.3
KBDistAI (with pruning)	91.8 ± 1.8	16.2 ± 3.7
Tiling-Pyramid	90.3 ± 1.8	23 ± 0.0
TopGen	90.9	42.1 ± 9.3
REGENT	91.8	70.1 ± 25.1

TABLE II
Results of **Promoters** dataset.

	Test %	*Size*
Rules alone	77.5 ± 5.0	–
KBDistAI (no pruning)	93.0 ± 2.8	12.2 ± 1.0
KBDistAI (with pruning)	95.5 ± 3.3	3.9 ± 2.3
Tiling-Pyramid	96.3 ± 1.8	34 ± 0.0
TopGen	94.8	40.2 ± 3.3
REGENT	95.8	74.9 ± 38.9

The time taken by both KBDistAI and Tiling-Pyramid is significantly less than that of the other approaches. KBDistAI and Tiling-Pyramid take a fraction of a minute to a few minutes of CPU time on each dataset used in the experiments. In contrast, TopGen and REGENT were reported to have taken several days to obtain the results reported in [25]. It is also worth noting that the networks generated by KBDistAI and Tiling-Pyramid are substantially more compact than those generated by TopGen and REGENT.

IV. Summary and Discussion

Theory refinement techniques offer an attractive approach to exploiting available domain knowledge to enhance the performance of data-driven knowledge acquisition systems. Neural networks have been used extensively in theory refinement systems that have been proposed in the literature. Most of such systems translate the domain theory into an initial neural network architecture and then train the network to refine the theory. The KBANN algorithm is demonstrated to outperform several other learning algorithms on some domains [12], [13]. However, a significant disadvantage of KBANN is its fixed network topology. TopGen and REGENT algorithms were proposed to eliminate this limitation and attempt to modify the network architecture. Experimental results demonstrate that TopGen and REGENT outperform KBANN on several applications. [24], [25].

Experimental results presented in this paper demonstrate that approaches to data-driven theory refinement using constructive neural network training algorithms such

as KBDistAl and Tiling-Pyramid are competitive in terms of of generalization accuracy with several of the more computationally expensive algorithms. Additional experiments are needed to conclusively determine the extent to which the incorporation of prior knowledge improves classification accuracy, and/or reduces learning time in real-world knowledge discovery applications. Examination of the changes to domain knowledge that are induced by training data is also of significant interest.

It can be argued that KBDistAl and Tiling-Pyramid are not *theory refinement* algorithms in a strict sense. They make use of the domain knowledge in inductive learning as opposed to truly *refining* the knowledge. Perhaps KBDistAl, Tiling-Pyramid, and related approaches should be more accurately described as a *knowledge guided* inductive theory construction systems.

There are several extensions and variants of KBDistAl that are worth exploring. Given the fact that DistAl relies on inter-pattern distances to induce classifiers from data, it is straightforward to extend it so as to handle a much broader class of problems including those that involve patterns of variable sizes (e.g., strings) or symbolic structures as long as suitable inter-pattern distance metrics can be defined. Some steps toward rigorous definitions of distance metrics based on information theory are outlined in [30]. Variants of DistAl and KBDistAl that utilize such distance metrics are currently under investigation.

Several authors have investigated approaches to rule extraction from neural networks in general, and connectionist theory refinement systems in particular [31], [32], [33]. One goal of such work is to represent the learned knowledge in a form that is comprehensible to humans. In this context, rule extraction from classifiers induced by KBDistAl and Tiling-Pyramid is of some interest.

Extensive experiments, on a wide range of knowledge discovery tasks in bioinformatics (e.g., protein structure prediction, identification of molecular structure-function relationships) using a broad range of data-driven knowledge refinement techniques (including constructive neural network approaches such as KBDistAl and Tiling-Pyramid, decision tree and rule induction techniques, Bayesian networks and related approaches) are needed in order to characterize the relative strengths and limitations of different techniques.

In several practical applications of interest, all of the data needed for synthesizing reasonably precise classifiers is not available at once. This calls for incremental algorithms that continually refine knowledge as more and more data becomes available. Computational efficiency considerations argue for the use of data-driven theory refinement systems as opposed to storing large volumes of data and rebuilding the entire classifier from scratch as new data becomes available. Some preliminary steps in this direction are described in [34].

A related problem involves knowledge discovery from large, physically distributed, dynamic data sources in a networked environment (e.g., data in genome databases). Given the large volumes of data involved, this argues for the use of data-driven theory refinement algorithms embedded in mobile software agents [34], [35] that travel from one data source to another, carrying with them, only the current knowledge base as opposed to approaches rely on shipping large volumes of data to a centralized repository where knowledge acquisition is performed. Thus, data-driven knowledge refinement algorithms constitute one of the key components of distributed knowledge network [34] environments for knowledge discovery in bioinformatics and related applications.

In several application domains, knowledge acquired on one task can often be utilized to accelerate knowledge acquisition on related tasks. Data-driven theory refinement is particularly attractive in applications that lend themselves to such cumulative multi-task learning [36]. The use of KBDistAl, Tiling-Pyramid, or similar algorithms in such scenarios remains to be explored.

References

[1] T. Mitchell, *Machine Learning*, McGraw Hill, New York, 1997.

[2] V. Honavar, R. Parekh, and J. Yang, "Machine learning: Principles and applications," in *Encyclopedia of Electrical and Electronics Engineering*, J. Webster, Ed. Wiley, New York, 1999, In press.

[3] M. Hassoun, *Fundamentals of Artificial Neural Networks*, MIT Press, Boston, MA, 1995.

[4] B. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, New York, 1996.

[5] J. W. Shavlik, "A framework for combining symbolic and neural learning," in *Artificial Intelligence and Neural Networks: Steps Toward Principled Integration*, V. Honavar and L. Uhr, Eds., pp. 561–580. Academic Press, New York, 1994.

[6] A. Ginsberg, "Theory reduction, theory revision, and retranslation," in *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston, MA, 1990, pp. 777–782, AAAI/MIT Press.

[7] D. Ourston and R. J. Mooney, "Theory refinement: Combining analytical and empirical methods," *Artificial Intelligence*, vol. 66, pp. 273–310, 1994.

[8] M. Kopel, R. Feldman, and A. Serge, "Bias-driven revision of logical domain theories," *Journal of Artificial Intelligence Research*, vol. 1, pp. 159–208, 1994.

[9] S. Donoho and L. Rendell, "Representing and restructuring domain theopries: A constructive induction approach," *Journal of Artificial Intelligence Research*, vol. 2, pp. 411–446, 1995.

[10] M. Pazzani and D. Kibler, "The utility of knowledge in inductive learning," *Machine Learning*, vol. 9, pp. 57–94, 1992.

[11] B. Richards and R. Mooney, "Automated refinement of first-order horn-clause domain theories," *Machine Learning*, vol. 19, pp. 95–131, 1995.

[12] G. Towell, J. Shavlik, and M. Noordwier, "Refinement of approximate domain theories by knowledge-based neural networks," in *Proceedings of the Eighth National Conference on Artificial Intelligence*, Boston, MA, 1990, pp. 861–866.

[13] G. Towell and J. Shavlik, "Knowledge-based artificial neural networks," *Artificial Intelligence*, vol. 70, no. 1–2, pp. 119–165, 1994.

[14] L. M. Fu, "Integration of neural heuristics into knowledge-based inference," *Connection Science*, vol. 1, pp. 325–340, 1989.

[15] B. F. Katz, "Ebl and sbl: A neural network synthesis," in *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, 1989, pp. 683–689.

[16] R. Parekh, J. Yang, and V. Honavar, "Constructive neural network learning algorithms for multi-category real-valued pattern classification," Tech. Rep. ISU-CS-TR97-06, Department of Computer Science, Iowa State University, 1997.

[17] V. Honavar, J. Yang, and R. Parekh, "Structural learning," in *Encyclopedia of Electrical and Electronics Engineering*, J. Webster, Ed. Wiley, New York, 1999, In press.

[18] J. Yang, R. Parekh, and V. Honavar, "DistAl: An inter-pattern distance-based constructive learning algorithm," in *Proceedings of the International Joint Conference on Neural Networks*, Anchorage, Alaska, 1998, pp. 2208–2213.

[19] J. Fletcher and Z. Obradović, "Combining prior symbolic knowledge and constructive neural network learning," *Connection Science*, vol. 5, no. 3,4, pp. 365–375, 1993.

[20] E. Baum and K. Lang, "Constructing hidden units using examples and queries," in *Advances in Neural Information Processing Systems, vol. 3*, R. Lippmann, J. Moody, and D. Touretzky, Eds., San Mateo, CA, 1991, pp. 904–910, Morgan Kaufmann.

[21] S. Gallant, "Perceptron based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 1, no. 2, pp. 179–191, June 1990.

[22] J. Mahoney and R. Mooney, "Comparing methods for refining certainty-factor rule-bases," in *Proceedings of the Eleventh International Conference on Machine Learning*, 1994, pp. 173–180.

[23] R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[24] D. W. Opitz and J. W. Shavlik, "Dynamically adding symbolically meaningful nodes to knowledge-based neural networks," *Knowledge-Based Systems*, vol. 8, no. 6, pp. 301–311, 1995.

[25] D. W. Opitz and J. W. Shavlik, "Connectionist theory refinement: Genetically searching the space of network topologies," *Journal of Artificial Intelligence Research*, vol. 6, pp. 177–209, 1997.

[26] J. Yang, R. Parekh, V. Honavar, and D. Dobbs, "Data-driven theory refinement using KBDistAl," in *Proceedings of the Third Symposium on Intelligent Data Analysis*, 1999, In press.

[27] J. Yang, R. Parekh, and V. Honavar, "DistAl: An inter-pattern distance-based constructive learning algorithm," *Intelligent Data Analysis*, 1999, To appear.

[28] R. Parekh and V. Honavar, "Constructive theory refinement in knowledge based neural networks," in *Proceedings of the International Joint Conference on Neural Networks*, Anchorage, Alaska, 1998, pp. 2318–2323.

[29] M. Frean, "A thermal perceptron learning rule," *Neural Computation*, vol. 4, pp. 946–957, 1992.

[30] D. Lin, "An information-theoretic definition of similarity," in *International Conference on Machine Learning*, 1998.

[31] G. Towell and J. Shavlik, "Extracting rules from knowledge-based neural networks," *Machine Learning*, vol. 13, pp. 71–101, 1993.

[32] L. M. Fu, "Knowledge based connectionism for revising domain theories," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 1, pp. 173–182, 1993.

[33] M. Craven, *Extracting Comprehensible Models from Trained Neural Networks*, Ph.D. thesis, Department of Computer Science, University of Wisconsin, Madison, WI, 1996.

[34] V. Honavar, L. Miller, and J. Wong, "Distributed knowledge networks," in *Proceedings of IEEE Information Technology Conference*, Piscataway, NJ, 1998, pp. 87–90.

[35] J. White, "Mobile agents," in *Software Agents*, J. Bradshaw, Ed. MIT Press, Cambridge, MA, 1997.

[36] S. Thrun, "Lifelong learning: A case study," Tech. Rep. CMU-CS-95-208, Carnegie Mellon University, Pittsburgh, PA, 1995.