REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY) 23-10-2017	2. REPORT TYPE Final Report	3. DATES COVERED (From - To) 6-Sep-2013 - 5-Mar-2017

4. TITLE AND SUBTITLE Final Report: 11.2 YIP Human In the Loop Statistical Relational Learners	5a. CONTRACT NUMBER W911NF-13-1-0432
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 611102
6. AUTHORS	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES Indiana University at Bloomington 509 East 3rd ST Bloomington, IN 47401 -3654	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSOR/MONITOR'S ACRONYM(S) ARO 11. SPONSOR/MONITOR'S REPORT NUMBER(S) 64577-NS-YIP.9

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Sriraam Natarajan
UU	UU	UU	UU		19b. TELEPHONE NUMBER 812-856-7615

RPPR Final Report
as of 23-Jan-2018

Agency Code:

Proposal Number: 64577NSYIP **Agreement Number: W911NF-13-1-0432**

INVESTIGATOR(S):

Name: Sriraam Natarajan
Email: natarasr@indiana.edu
Phone Number: 8128567615
Principal: Y

Organization: **Indiana University at Bloomington**
Address: 509 East 3rd ST, Bloomington, IN 474013654
Country: USA
DUNS Number: 006046700 EIN: 356001673
Report Date: 05-Jun-2017 Date Received: 23-Oct-2017
Final Report for Period Beginning 06-Sep-2013 and Ending 05-Mar-2017
Title: 11.2 YIP Human In the Loop Statistical Relational Learners
Begin Performance Period: 06-Sep-2013 **End Performance Period:** 05-Mar-2017
Report Term: 0-Other
Submitted By: Sriraam Natarajan Email: natarasr@indiana.edu
Phone: (812) 856-7615
Distribution Statement: 1-Approved for public release; distribution is unlimited.

STEM Degrees: 1 **STEM Participants:** 2

Major Goals: 1. Incorporate Human advice to sequential decision-making tasks
2. Allow for probabilistic logic models to accept and solicit applications as needed.
3. Allow for richer forms of interaction between the agent and the human.

Accomplishments: 1. Advice as preferences over actions/labels
1.a Sequential decision-making framework
1.b Boosted relational learning framework

2. Advice as additional information over examples
2.a Privileged framework with decision trees

3. Applications
3.a Drone flying, traffic control
3.b Image segment labeling
3.c Extracting information from PubMed
3.d Augmeting knowledge to Never Ending Language Learner (NELL)

Published 8 papers in peer-reviewed conferences, 2 papers at premier workshops at these conferences, 1 Machine Learning Journal Paper and 2 Books.

Details of the different accomplishments are presented in the document uploaded.

Training Opportunities: Phillip Odom, graduate student was fully funded from this grant and he has finished his PhD in July 2017.

He is currently a Research Scientist in Georgia Tech.

Results Dissemination: The resulting research was presented in a variety of top-tier conferences and journals including International Conference on AI (AAAI), International Conference on Data Mining (ICDM), European Conference on Machine Learning (ECML), AI in Medicine (AIME), International Conference on Inductive Logic Programing (ILP), Machine Learning Journal (MLJ) and finally as two books - Springer series and Morgan and Claypool book series.

Honors and Awards: Best Student Paper Award - BeyondLabeler Workshop at IJCAI 2015

Protocol Activity Status:

Technology Transfer: Nothing to Report

CONFERENCE PAPERS:

Publication Type: Conference Paper or Presentation **Publication Status:** 1-Published
Conference Name: European Conference in Machine Learning
Date Received: 23-Oct-2017 Conference Date: 19-Sep-2016 Date Published:
Conference Location: RIVA DEL GARDA
Paper Title: Actively Interacting with Experts: A Probabilistic Logic Approach
Authors: Phillip Odom, Sriraam Natarajan
Acknowledged Federal Support: **Y**

Publication Type: Conference Paper or Presentation **Publication Status:** 0-Other
Conference Name: International Conference on Inductive Logic Programming
Date Received: 23-Oct-2017 Conference Date: 04-Sep-2016 Date Published:
Conference Location: London
Paper Title: Learning through Advice-Seeking via Transfer
Authors: Phillip Odom, Raksha Kumaraswamy, Kristian Kersting, Sriraam Natarajan
Acknowledged Federal Support: **Y**

Publication Type: Conference Paper or Presentation **Publication Status:** 0-Other
Conference Name: 2015 IEEE International Conference on Data Mining (ICDM)
Date Received: 23-Oct-2017 Conference Date: 14-Nov-2015 Date Published:
Conference Location: Atlantic City, NJ, USA
Paper Title: Transfer Learning via Relational Type Matching
Authors: Raksha Kumaraswamy, Phillip Odom, Kristian Kersting, David Leake, Sriraam Natarajan
Acknowledged Federal Support: **Y**

Fianl Report for Human-in-the-Loop Statistical Relational Learners

We have made progress in several directions for Human-in-the-Loop learning including 1) building algorithms capable of incorporating human advice [4, 7, 5, 12, 6], and 2) allowing algorithms to actively acquire useful advice by querying human domain experts [9, 10, 8, 11]. Our algorithms are more robust to noisy or missing training data, where standard machine learning algorithms may struggle. We have developed and applied advice-based methods to different learning formalisms including inverse reinforcement learning [4] and statistical relational learning [7, 5, 8]. We have also applied our algorithms in complex tasks such as natural language processing for extracting adverse drugs events from text [5].

First we will describe the advice-based methods that we have developed. Then we talk about how advice-based learning can be extended for actively seeking advice, allowing the learning algorithm to acquire targeted advice.

1 Key Research Accomplishments

1.1 Advice-based Learning

We have developed two different methods for receiving advice from the domain expert including label preferences and privileged features. Label preferences allow the expert to describe some of the characteristics that make one prediction more likely than another, e.g., "Yellow traffic signals mean *slow down* and not *speed up*". Privileged features allow the expert to give additional information to the learning algorithm about the training examples.

1.1.1 Advice as Label Preferences

The goal of our human advice as label preferences [7, 5] is to target (and correct) any errors in the training data. In this way, our framework allows for the learning of a more accurate and robust model. The advice is specified by the expert naturally in first-order logic. This allows for non-machine learning experts to provide valuable advice. The form of our advice is inspired by preference elicitation [1] where the expert can provide information on what labels the model should prefer for a set of examples. The expert provides label preferences in the form of first-order logic horn clauses.

Formally, our advice is defined as,

Definition 1 *Relational advice set (RAS), R is specified as a set of relational advice rules (RAR),* $r_1, r_2, \ldots, r_A$. *Each RAR,* r_a *is defined using the triple*

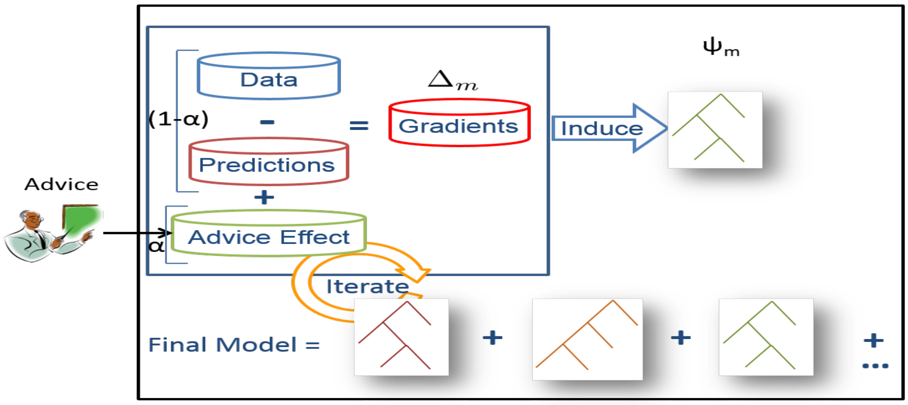


Figure 1: Our advice-based boosting framework.

$\langle F, l+, l-\rangle$ *where* F *is the relational advice constraint (RAC) clause specifying the subset of examples,* $l+$ *is the preferred label and* $l-$ *is the avoided label.*

Definition 2 *A relational advice constraint (RAC),* F *is defined using a Horn clause* $\wedge_i f_i(\mathbf{x_i}) \Rightarrow label(\mathbf{x_e})$*, where* $\wedge_i f_i(\mathbf{x_i})$ *specifies the conjunction of conditions under which the advice applies on the example arguments* $\mathbf{x_e}$*.*

Intuitively, the goal of learning with label preferences is to learn a model that predicts a higher chance of preferred labels than avoided labels. If $\mathbf{s}$ represents the set of training examples for which the advice is applicable, then the learned model should have a higher probability of the preferred target than the probability of the avoided target in the training examples, i.e., $\forall s_i \in \mathbf{s}, P(l+(s_i)) \geq P(l-(s_i))$.

Our overall approach is shown in Figure 1. Inspired by prior work of Gimpel and Smith [2], we introduce a cost function in the denominator of the log-likelihood function. While they employ the use of a regularization term for a log-linear model, we employ this as a penalty term for violating the advice provided by the expert. This modified log-likelihood (MLL) function using the functional representation is given as,

$$MLL(\mathbf{x}, \mathbf{y}) = \sum_{x_i \in \mathbf{x}} \log \frac{\exp\left(\psi(x_i; y_i)\right)}{\sum_{y'} \exp\left(\psi(x_i; y') + cost(y_i, y', \psi)\right)} \tag{1}$$

Our cost function is used to penalize the model that does not fit to the advice. Since the cost function penalty depends only on the advice and the current model, and not on the example labels y and y'; we can redefine it as $cost(x_i, \psi)$. We define the cost function as

$$cost(x_i, \psi) = -\lambda \times \psi(x_i) \times [n_t(x_i) - n_f(x_i)] \tag{2}$$

We use n_t to indicate the number of advice rules that prefer the example to be true and n_f to be the number of rules that prefer it to be false. We use λ to scale the cost function and $\psi(x_i)$ is the current value of the ψ function for the x_i.

We now use functional-gradient boosting to maximize our modified objective function (MLL). The cost function in Equation 1 can be replaced with Equation 2 and it becomes:

$$MLL(\mathbf{x}) = \sum_i \log P(y_i, x_i; \psi) + \lambda \cdot \psi(x_i) \cdot [n_t(x_i) - n_f(x_i)]$$

The gradients for the MLL can be similarly defined:

$$\Delta(x_i) \quad = \quad I(y_i = 1) - P(y_i = 1; \psi) + \lambda \cdot [n_t(x_i) - n_f(x_i)]$$

Intuitively when the example label is the preferred target in more advice models than the avoided target, $n_t(x_i) - n_f(x_i)$ is set to be positive. This will result in pushing the gradient of these examples in the positive direction (towards $+\infty$). Conversely when the example label should be avoided in more advice models, $n_t(x_i) - n_f(x_i)$ is set to be negative which will result in pushing the gradient of this example in the negative direction (towards $-\infty$). Examples where the advice does not apply or has equally contradictory advice, $n_t(x_i) - n_f(x_i)$ is 0. Hence, our approach can also handle *conflicting advice for the same example.*

The strength of our approach not only comes from the concise way that our method is able to handle advice, but also that this advice is used to improve the model throughout the learning process. This framework is versatile and can be applied to classification, policy learning and used to transfer information across domains (the advice captures the difference between the domains).

1.1.2 Advice as Privileged Information

The key idea behind advice as privileged information is to use the privileged features – available only at training time – to guide the search in the normal features in order to learn a better model [14, 15]. Privileged information provides an alternate method for human domain experts to express their knowledge. The intuition of our approach for learning with privileged information is to bias the learning algorithm to make similar predictions in both features spaces [12]. Thus, making it more likely that similar examples in the privileged space will have similar predictions over the normal features, potentially improving generalization. More formally, the problem is defined as:

Given: A set of training examples $[\langle y_i, \mathbf{x}_i^{\mathbf{CF}}, \mathbf{x}_i^{\mathbf{PF}} \rangle]$ and a set of test examples $[\langle y_i, \mathbf{x}_i^{\mathbf{CF}} \rangle]$, where $\mathbf{F} = \mathbf{CF} \cup \mathbf{PF} \quad \& \quad \mathbf{CF} \cap \mathbf{PF} = \emptyset$

To do: Learn a classifier that employs only the classifier features **CF** for classifying the test data and can utilize the privileged features **PF** effectively in learning a better model.

F is the set of all features, **CF** is the set of features that are available at both training and test time (*classifier or normal features*), **PF** are the privileged features, y_i is the label and $\mathbf{x}_i$ are the features of the i^{th} example. [] denotes sets. The input to the algorithm is the set of all examples $[\langle y_i, \mathbf{x}_i^{\mathbf{CF}}, \mathbf{x}_i^{\mathbf{PF}} \rangle]$.

While we explored several different assumptions about quality of human expertise, we describe the setting where the expert is not assumed to be optimal. Thus, the goal is to learn a model that minimizes the error of the model over the training labels and the margin between the distributions $P_D(y|\mathbf{x}^{\mathbf{PF}})$ and $P(y|\mathbf{x}^{\mathbf{CF}})$,

$$\min \sum_i [-\log(P(y_i|\mathbf{x}_i^{\mathbf{CF}})) + \alpha \cdot KL(P_D(y_i|\mathbf{x}_i^{\mathbf{PF}})||P(y_i|\mathbf{x}_i^{\mathbf{CF}}))]$$

where $-\sum_i \log(P(y_i|\mathbf{x}_i^{\mathbf{CF}})$, the negative log-likelihood of the training data, is used to model the error and KL denotes the KL-divergence between P_D and P given by $\sum_i P_D(i) \log \frac{P_D(i)}{P(i)}$. We use α to model the trade-off between fitting to the labeled data versus fitting to the distribution learned over **PF**. We can now use gradient boosting w.r.t $\psi(\mathbf{x}^{\mathbf{CF}})$.

The standard gradient is combined with the gradient from the KL-divergence to compute the final gradient for each example:

$$\Delta(\mathbf{x}_i^{\mathbf{CF}}) = I(y_i = 1) - P(y_i = 1|\mathbf{x}_i^{\mathbf{CF}}) - \alpha \cdot \left(P(y_i = 1|\mathbf{x}_i^{\mathbf{CF}}) - P_D(y_i = 1|\mathbf{x}_i^{\mathbf{PF}})\right)$$

Intuitively, if the learned distribution has a higher probability of an example belonging to the positive class compared to the **PF** distribution, $P(y_i = 1|\mathbf{x}_i^{\mathbf{CF}}) - P_D(y_i = 1|\mathbf{x}_i^{\mathbf{PF}})$ would be positive and the gradient would be pushed lower. Hence the additional term would push the gradient (weighted by α) towards the true distribution as predicted by **PF**. Note that gradients for the privileged model can be found by switching P and P_D.

The parameter[1], α controls the influence of the privileged data on the learned distribution. In the extreme case of $\alpha = 0$, **PF** are completely ignored, i.e., **PF** is obtained from a noisy expert and we end up with the standard functional gradient. While it is potentially possible to choose α via some experimental method such as cross-validation, given the intuition that it represents the importance of privileged information, we define α for the classifier model as,

$$\alpha \propto \frac{\sum_i I(P_D(y_i = y_i^*|\mathbf{PF}) >= P(y_i = y_i^*|\mathbf{CF}))}{N}$$

which is proportional to the fraction of the number of examples where the privileged features yield a better distribution over the observed labels (y^*) than the classifier features. As a better classifier model is learned, the value of α goes down giving more importance to the data.

Unlike label preferences which may focus on explanations for general trends or special exceptions, privileged information can provide additional context for the algorithm to utilize during learning. Our α selection also allows the algorithm to adapt to the varying levels of expertise of the human advice-giver.

1.2 Active Advice Seeking

The goal of active advice seeking is to generate queries to the expert. This is similar to active learning [13] which solicits examples for an expert to label. In this way active learning builds up a useful dataset while having as few examples labeled as possible. The key difference between active learning and active advice seeking is that active advice seeking provides a much more expressive medium for communication. A single piece of advice could cover many ground examples. There are many more possible advice than ground states so the key challenge in

---

[1] Note that this α fulfills a similar purpose as the one introduced for label preferences.

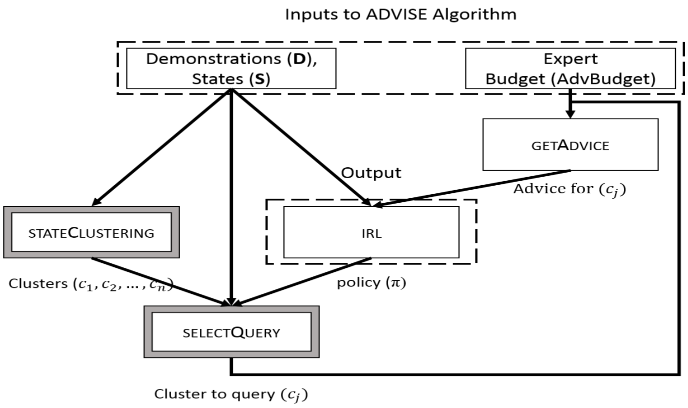


Figure 2: Active Advice Seeking for Inverse Reinforcement Learning.

active advice seeking is in selecting the appropriate advice at each step. We will now briefly discuss active advice seeking in the context of sequential decision-making problems and relational learning tasks.

1.2.1 Sequential Decision-Making

Our previous work on advice for inverse reinforcement learning (IRL) defined advice as action preferences [4]. Experts could specify rules like "Brake when approaching red lights (brake lights or stoplights), do not accelerate". Internally, the learned policy should penalize the model for accelerating in these states and ensure that braking is more likely. Therefore, advice consists of 1) a cluster of states, 2) a set of actions which are preferred and 3) a set of actions which are avoided. We previously showed that advice in this form can be useful when learning [4]. Note that the propositional representation has no explicit way of representing "approaching red lights", but the algorithm could cluster all such states together.

The overall goal of ADVice SEeking (ADVISE) is to define the cluster of states of interest. Once this cluster is known, the expert can be queried to provide both the set of preferred as well as avoided labels. Our approach, shown in Figure 2, takes as input a description of the state space, the demonstrations (training examples) and query budget. It selects the advice clusters in two steps. First, we perform clustering on the state space based on both the action distribution specified by the demonstrations and the description of the state space. This should cluster states for which the demonstrations suggest similar actions and which have similar state descriptions. The clusters generated represent potential queries. In the second step, the algorithm ranks the clusters based on their uncertainty.

IRL has two sources of uncertainty. (1) The inputs (demonstrations) can be ambiguous in some states about the correct actions due to either too few demonstrations in those states or through mistakes– incorrect actions. (2) The other source of uncertainty is that the output (final policy) may not clearly delineate an action for the agent. Therefore, we define the uncertainty of a state as a combination of these two types of uncertainty.

We define demonstration uncertainty ($F(s_i)$) to be the entropy of the distribution of actions selected by the expert at state s_i. This uncertainty is fixed as we typically assume that the set of demonstrations do not change. IRL learns a reward function which we use to generate a distribution over action for every state. Uncertainty with respect to the current policy ($G(s_i)$) is the entropy of the action distribution for each state.

Definition 3 *State-level uncertainty* $U_s(s_i)$ *is given by* $U(s_i) = w_p F(s_i) + (1 - w_p) G(s_i)$ *where* $F(s_i)$ *is the uncertainty w.r.t the number of demonstrations provided for state* s_i, w_p *is the corresponding weight and* $G(s_i)$ *is the uncertainty w.r.t the policy learned for state* s_i.

Uncertainty at the cluster level is a function of the uncertainty of each state assigned to that cluster.

Definition 4 *The cluster-level uncertainty* U_c *is given by* $U_c(\boldsymbol{c}_j) = \frac{\sum_{s_i \in \boldsymbol{c}_j} U_s(s_i)}{|\boldsymbol{c}_j|}$.

The iterative algorithm queries advice from the expert, updates its score for the rest of the clusters and repeats until it has exhausted its query budget. We have achieved impressive results in many standard reinforcement learning tasks [9]. While comparing favorably to standard active learning, the key drawback of this approach is that the advice is restricted to a propositional representation. We extend this work to the relational setting.

1.2.2 Relational Learning

Advice in relational tasks is specified as label preference similar to those used for inverse reinforcement learning. However, the sets of states to which the advice is applied can be described by a clause in first-order logic. Formalizing the advice in this way makes queries more interpretable as the clause represents a compact description of the examples.

The query in active advice seeking for relational domains is a relational advice constraint (RAC), Definition 2, which describes the area of the state space about which the algorithm is requesting knowledge. We define the uncertainty of an example to be the entropy of the current prediction of that example. The key insight to our work is that we can learn a tree that captures both the query as well as the estimated uncertainty. We fit a relational regression tree to the uncertainty values of the current model. This tree can be written as a weighted list of first-order logic clauses. The clause represents the query while the weight represents the estimated uncertainty of that cluster. Our active advice seeking algorithm for relational domains is shown in Figure 3.

Active advice seeking increases the impact and usability of our advice-based algorithm by discovering and correcting uncertainties in the current model. While we do not explicitly minimize the number of queries, our algorithm increases the impact of each interaction with the expert by pointing them towards particular regions of the state space. We have successfully applied our approach to standard relational tasks [10].

1.2.3 Advice-Seeking via Transfer Learning

In our previous work, queries were generated based on the current model learned from the training data. While effective, this ignores any knowledge that may

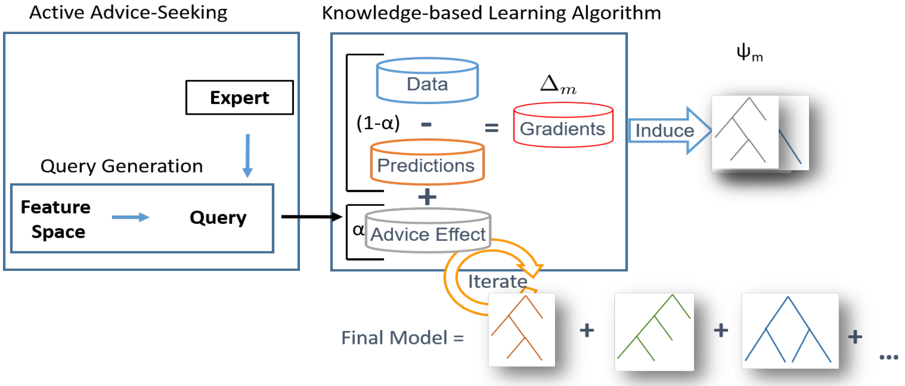
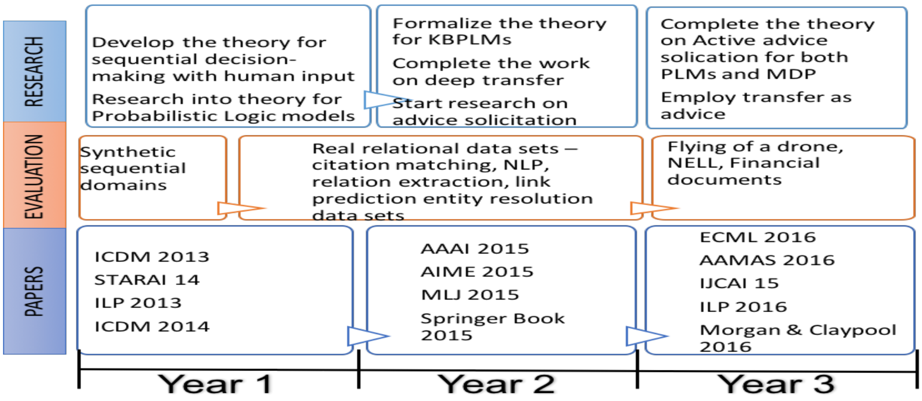


Figure 3: Active Advice Seeking for Relational Models.



	Year 1	Year 2	Year 3
RESEARCH	Develop the theory for sequential decision-making with human input Research into theory for Probabilistic Logic models	Formalize the theory for KBPLMs Complete the work on deep transfer Start research on advice solicitation	Complete the theory on Active advice solication for both PLMs and MDP Employ transfer as advice
EVALUATION	Synthetic sequential domains	Real relational data sets – citation matching, NLP, relation extraction, link prediction entity resolution data sets	Flying of a drone, NELL, Financial documents
PAPERS	ICDM 2013 STARAI 14 ILP 2013 ICDM 2014	AAAI 2015 AIME 2015 MLJ 2015 Springer Book 2015	ECML 2016 AAMAS 2016 IJCAI 15 ILP 2016 Morgan & Claypool 2016

Figure 4: Research Timeline from YIP Award.

exists about other domains. Transfer learning aims to use knowledge in one domain to improve performance in another task. Such transferred knowledge could be effective advice in the domain of interest.

We utilize Language-bias Transfer Learning [3] to transfer first-order logic Horn clauses from a different domain. It constructs clauses in a target domain that have similar relational structure as clauses given in the target domain. These clauses can be used directly by our active advice seeking algorithm described earlier. We were able to show that this transfer procedure was capable of generating reasonable advice across several different relational datasets [8].

2 Research Timeline

Figure 4 presents the progress of the research in the grant. We have organize this timeline into three specific aspects - the research direction, the evaluation platforms and data sets and finally the papers/books that are the result of the research. Our initial focus was on sequential decision making models and then we moved on to the probabilistic logic models. We closed the loop by making the agent learn to know what it knows and solicit information about what it does not know. We performed rigorous evaluations on several datasets some of which were synthetic and created by us. But our evaluations also included several real data sets such as microscopic images of fruit flies, mining information from PubMed abstracts, citation matching from large citeseer data sets, the Never Ending Language Learner (NELL) from CMU and flying a drone. The resulting papers appeared in a variety of conferences and journals including International Conference on AI (AAAI), International Conference on Data Mining (ICDM), European Conference on Machine Learning (ECML), AI in Medicine (AIME), International Conference on Inductive Logic Programing (ILP), Machine Learning Journal (MLJ) and finally as two books - Springer series and Morgan and Claypool book series.

3 Next Steps

Research in human-in-the-loop learning has primarily focused on a single expert as the source of knowledge. In many domains such as healthcare, there is a large body of experts which may have differing levels of expertise or areas of specialty. Similarly, learners should be able to utilize knowledge from multiple experts and seek advice based on the level of expertise or specialization. This may involve explicitly reasoning about the quality of the available experts.

Furthermore, human experts may not be the only relevant source of information. Many knowledge sources exist on the internet such as Wikipedia or PubMed where information extraction techniques could generate useful knowledge. While this knowledge noisy depending on the source of information, we can reason about it explicitly. Another potential source of information could be related tasks. Transfer learning focuses on using knowledge from one domain to improve performance in another. If HIL algorithms are able to incorporate knowledge from other sources directly in areas where it is available, it would allow for more effective use the human expert. HIL algorithms require continuous interaction with human experts and would benefit from a more refined interface and more expressive interactions with expert. Incorporating more expressive interaction, effective methods of providing useful information and suggesting feedback to the human expert would strengthen the impact of human-in-the-loop systems.

References

[1] Craig Boutilier. A pomdp formulation of preference elicitation problems. In *AAAI*, 2002.

[2] Kevin Gimpel and Noah A. Smith. Softmax-margin crfs: Training log-linear models with cost functions. In *HLT-NAACL*, 2010.

[3] Raksha Kumaraswamy, Phillip Odom, Kristian Kersting, David Leake, and Sriraam Natarajan. Transfer learning across relational and uncertain domains: A language-bias approach. In *ICDM (Under Review)*, 2015.

[4] Gautam Kunapuli, Phillip Odom, Jude Shavlik, and Sriraam Natarajan. Guiding autonomous agents to better behaviors through human advice. In *ICDM*, 2013.

[5] Phillip Odom, Vishal Bangera, Tushar Khot, David Page, and Sriraam Natarajan. Extracting adverse drug events from text using human advice. In *AIME*, 2015.

[6] Phillip Odom, Tushar Khot, and Sriraam Natarajan. Learning probabilistic logic models with human advice. In *AAAI Spring Symposium on Knowledge Representation and Reasoning*, 2015.

[7] Phillip Odom, Tushar Khot, Reid Porter, and Sriraam Natarajan. Knowledge-based probabilistic logic learning. In *AAAI*, 2015.

[8] Phillip Odom, Raksha Kumaraswamy, Kristian Kersting, and Sriraam Natarajan. Learning through advice-seeking via transfer. In *ILP*, 2016.

[9] Phillip Odom and Sriraam Natarajan. Active advice seeking for inverse reinforcement learning. In *AAMAS*, 2016.

[10] Phillip Odom and Sriraam Natarajan. Actively interacting with experts: A probabilistic logic approach. In *ECMLPKDD*, 2016.

[11] Phillip Odom and Sriraam Natarajan. Probabilistic logic learning via active advice seeking. In *First Workshop on Human is More Than a Labeler (BeyondLabeler)*, 2016.

[12] Rahul Pasunuri, Phillip Odom, Tushar Khot, Kristian Kersting, and Sriraam Natarajan. Learning with privileged information: Decision-trees and boosting. In *First Workshop on Human is More Than a Labeler (BeyondLabeler)*, 2016.

[13] Burr Settles. *Active Learning.* Morgan & Claypool, 2012.

[14] V Vapnik and A Vashist. A new learning paradigm: Learning using privileged information. In *Neural Networks*, volume 22, pages 544–557, 2009.

[15] Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: Similarity control and knowledge transfer. *JMLR*, 16:2023–2049, 2015.