Active Learning of Strict Partial Orders: A Case Study on Concept Prerequisite Relations

Chen Liang[1] Jianbo Ye[1] Han Zhao[2] Bart Pursel[1] C. Lee Giles[1]
[1]The Pennsylvania State University
[2]Carnegie Mellon University
{cul226, jxy198, bkp10, clg20}@psu.edu
han.zhao@cs.cmu.edu

ABSTRACT

Strict partial order is a mathematical structure commonly seen in relational data. One obstacle to extracting such type of relations at scale is the lack of large scale labels for building effective data-driven solutions. We develop an active learning framework for mining such relations subject to a strict order. Our approach incorporates relational reasoning not only in finding new unlabeled pairs whose labels can be deduced from an existing label set, but also in devising new query strategies that consider the relational structure of labels. Our experiments on concept prerequisite relations show our proposed framework can substantially improve the classification performance with the same query budget compared to other baseline approaches.

1. INTRODUCTION

Pool-based active learning is a learning framework where the learning algorithm is allowed to access a set of unlabeled examples and ask for the labels of any of these examples [3]. Its goal is to learn a good classifier with significantly fewer labels by actively directing the queries to the most "valuable" examples. In a typical setup of active learning, the label dependency among labeled or unlabeled examples is not considered. But data and knowledge in the real world are often embodied with prior relational structures. Taking into consideration those structures in building machine learning solutions can be necessary and crucial. The goal of this paper is to investigate the query strategies in active learning of a strict partial order, namely, when the ground-truth labels of examples constitute an irreflexive and transitive relation. In this paper, we develop efficient and effective algorithms extending popular query strategies used in active learning to work with such relational data. We study the following problem in the active learning context:

Problem. Given a finite set V, a strict order on V is a type of irreflexive and transitive (pairwise) relation. Such a strict order is represented by a subset $G \subseteq V \times V$. Given an unknown strict order G, an oracle W that returns $W(u,v) = -1 + 2 \cdot \mathbf{1}[(u,v) \in G] \in \{-1, 1\}$, and a feature extractor $\mathcal{F} : V \times V \mapsto \mathbb{R}^d$, find $h : \mathbb{R}^d \mapsto \{-1, 1\}$ from a hypothesis class $\mathcal{H}$ that predicts whether or not $(u,v) \in G$ for each pair $(u,v) \in V \times V$ and $u \neq v$ (using $h(\mathcal{F}(u,v))$) by querying W a finite number of (u,v) pairs from $V \times V$.

Our main focus is to develop reasonable query strategies in active learning of a strict order exploiting both the knowledge from (non-consistent) classifiers trained on a limited number of labeled examples and the deductive structures among pairwise relations. Our work also has a particular focus on *partial orders*. If the strict order is total, a large school called "learning to rank" has studied this topic [10], some of which are under the active learning setting [4]. Learning to rank relies on binary classifiers or probabilistic models that are consistent with the rule of a total order. Such approaches are however limited in a sense to principally modeling a partial order: a classifier consistent with a total order will always have a non-zero lower bound of error rate, if the ground-truth is a partial order but not a total order.

In our active learning problem, incorporating the deductive relations of a strict order in soliciting examples to be labeled is non-trivial and important. The challenges motivating us to pursue this direction can be explained in three folds: First, any example whose label can be deterministically reasoned from a labeled set by using the properties of strict orders does not need further manual labeling or statistical prediction. Second, probabilistic inference of labels based on the independence hypothesis, as is done in the conventional classifier training, is *not* proper any more because the deductive relations make the labels of examples dependent on each other. Third, in order to quantify how valuable an example is for querying, one has to combine uncertainty and logic to build proper representations. Sound and efficient heuristics with empirical success are to be explored.

One related active learning work that deals with a similar setting to ours is [13], whereas equivalence relations are considered instead. Particularly, they made several crude approximations in order to expedite the expected error calculation to a computational tractable level. We approach the design of query strategies from a different perspective while keeping efficiency as one of our central concerns.

To empirically study the proposed active learning algorithm, we apply it to *concept prerequisite learning problem* [15, 8], where the goal is to predict whether a concept A is a pre-

requisite of a concept B given the pair (A, B). Although there have been some research efforts towards learning prerequisites [16, 15, 8, 17], the mathematical nature of the prerequisite relation as strict partial orders has not been investigated. In addition, one obstacle for effective learning-based solutions to this problem is the lack of large scale prerequisite labels. Liang et al. [9] applied standard active learning to this problem without utilizing relation properties of prerequisites. Active learning methods tailored for strict partial orders provide a good opportunity to tackle the current challenges of concept prerequisite learning.

Our main contributions are summarized as follows: Fist, we propose a new efficient reasoning module for monotonically calculating the deductive closure under the assumption of a strict order. This computational module can be useful for general AI solutions that need fast reasoning in regard to strict orders. Second, we apply our reasoning module to extend two popular active learning approaches to handle relational data and empirically achieve substantial improvements. This is the first attempt to design active learning query strategies tailored for strict partial orders. Third, under the proposed framework, we solve the problem of concept prerequisite learning and our approach appears to be successful on data from four educational domains, whereas previous work have not exploited the relational structure of prerequisites as strict partial orders in a principled way.

2. REASONING OF A STRICT ORDER

2.1 Preliminary

DEFINITION 1 (STRICT ORDER). *Given a finite set V, a subset G of $V \times V$ is called a strict order if and only if it satisfies the two conditions: (i) if $(a, b) \in G$ and $(b, c) \in G$, then $(a, c) \in G$; (ii) if $(a, b) \in G$, then $(b, a) \notin G$.*

DEFINITION 2 (G-ORACLE). *For two subsets $G, H \subseteq V \times V$, a function denoted as $W_H(\cdot, \cdot) : H \mapsto \{-1, 1\}$ is called a G-oracle on H iff for any $(u, v) \in H$, $W_H(u, v) = -1 + 2 \cdot \mathbf{1}[(u, v) \in G]$.*

The G-oracle returns a label denoting whether a pair belongs to G.

DEFINITION 3 (COMPLETENESS OF AN ORACLE). *A G-oracle of strict order W_H is called complete if and only if H satisfies: for any $a, b, c \in V$, (i) if $(a, b) \in H \cap G$, $(b, c) \in H \cap G$, then $(a, c) \in H \cap G$; (ii) if $(a, b) \in H \cap G$, $(a, c) \in H \cap G^c$, then $(b, c) \in H \cap G^c$; (iii) if $(b, c) \in H \cap G$, $(a, c) \in H \cap G^c$, then $(a, b) \in H \cap G^c$; (iv) if $(a, b) \in H \cap G$, then $(b, a) \in H \cap G^c$, where G^c is the complement of G.*
W_H is called complete if it is consistent under transitivity when restricted on pairs from H.

DEFINITION 4 (CLOSURE). *Given a strict order G, for any $H \subseteq V \times V$, its closure is defined to be the smallest set $\overline{H}$ such that $H \subseteq \overline{H}$ and the G-oracle $W_{\overline{H}}$ is complete.*

DEFINITION 5 (DESCENDANT AND ANCESTOR). *Given a strict order G of V and $a \in V$, its ancestor subject to G is $A_a^G := \{b \mid (b, a) \in G\}$ and its descendant is $D_a^G := \{b \mid (a, b) \in G\}$.*

2.2 Reasoning Module for Closure Calculation

With the definitions in the previous section, this section proposes a reasoning module that is designed to monotonically calculate the deductive closure for strict orders. Remark that a key difference between the traditional transitive closure and our definition of closure (Definition 3&4) is that the former only focuses on G but the latter requires calculation for both G and G^c. In the context of machine learning, relations in G and G^c correspond to positive examples and negative examples, respectively. Since both of these examples are crucial for training classifiers, existing algorithms for calculating transitive closure such as the Warshall algorithm are not applicable. Thus we propose the following theorem for monotonically computing the closure. Please refer to supplemental material for the proofs.

THEOREM 1. *Let G be a strict order of V and W_H a complete G-oracle on $H \subseteq V \times V$. For any pair $(a, b) \in V \times V$, define the notation $C_{(a,b)}$ by*

(i) If $(a, b) \in H$, $C_{(a,b)} := H$.

(ii) If $(a, b) \in G^c \cap H^c$, $C_{(a,b)} := H \cup N'_{(a,b)}$ where

$$N'_{(a,b)} := \{(d, c) | c \in A_b^{G \cap H} \cup \{b\}, d \in D_a^{G \cap H} \cup \{a\}\},$$

and particularly $N'_{(a,b)} \subseteq G^c$.

(iii) If $(a, b) \in G \cap H^c$, $C_{(a,b)} := H \cup N_{(a,b)} \cup R_{(a,b)} \cup S_{(a,b)} \cup T_{(a,b)} \cup O_{(a,b)}$, where

$$\begin{aligned}
N_{(a,b)} &:= \{(c, d) \mid c \in A_a^{G \cap H} \cup \{a\}, d \in D_b^{G \cap H} \cup \{b\}\},\\
R_{(a,b)} &:= \{(d, c) \mid (c, d) \in N_{(a,b)}\},\\
S_{(a,b)} &:= \{(d, e) \mid c \in A_a^{G \cap H} \cup \{a\}, d \in D_b^{G \cap H} \cup \{b\},\\
&\qquad (c, e) \in G^c \cap H\},\\
T_{(a,b)} &:= \{(e, c) \mid c \in A_a^{G \cap H} \cup \{a\}, d \in D_b^{G \cap H} \cup \{b\},\\
&\qquad (e, d) \in G^c \cap H\},\\
O_{(a,b)} &:= \bigcup\nolimits_{(c,d) \in S_{(a,b)} \cup T_{(a,b)}} N''_{(c,d)},\\
N''_{(c,d)} &:= \{(f, e) \mid e \in A_d^{G \cap (H \cup N_{(a,b)})} \cup \{d\},\\
&\qquad f \in D_c^{G \cap (H \cup N_{(a,b)})} \cup \{c\}\}.
\end{aligned}$$

In particular, $N_{(a,b)} \subseteq G$ and $R_{(a,b)} \cup S_{(a,b)} \cup T_{(a,b)} \cup O_{(a,b)} \subseteq G^c$.

For any pair $(x, y) \in V \times V$, the closure of $H' = H \cup \{(x, y)\}$ is $C_{(x,y)}$.

Figure 1 provides an informal explanation of each necessary condition (except for $R_{(a,b)}$) mentioned in the theorem. If (a, b) is a positive example, i.e. $(a, b) \in G$, then (i) $N_{(a,b)}$ is a set of inferred positive examples by transitivity; (ii) $R_{(a,b)}$ is a set of inferred negative examples by irreflexivity; (iii) $S_{(a,b)}$ and $T_{(a,b)}$ are sets of inferred negative examples by transitivity; (iv) $O_{(a,b)}$ is a set of negative examples inferred from $S_{(a,b)}$ and $T_{(a,b)}$. If (a, b) is a negative example, i.e. $(a, b) \in G^c$, then $N'_{(a,b)}$ is a set of negative examples inferred by transitivity.

3. POOL-BASED ACTIVE LEARNING

The pool-based sampling [7] is a typical active learning scenario in which one maintains a labeled set $\mathcal{D}_l$ and an unlabeled set $\mathcal{D}_u$. In particular, we let $\mathcal{D}_u \cup \mathcal{D}_l = \mathcal{D} = \{1, \ldots, n\}$ and $\mathcal{D}_u \cap \mathcal{D}_l = \varnothing$. For $i \in \{1, \ldots, n\}$, we use $\mathbf{x}_i \in \mathbb{R}^d$ to denote a feature vector representing the i-th instance, and $y_i \in \{-1, +1\}$ to denote its groundtruth class label. At each round, one or more instances are selected from $\mathcal{D}_u$ whose label(s) are then requested, and the labeled instance(s) are

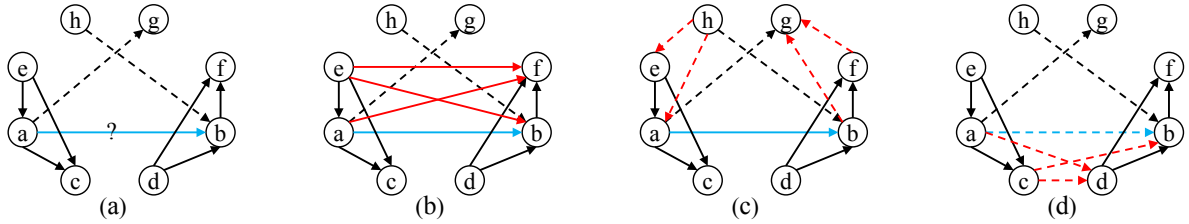


Figure 1: Following the notations in Theorem 1: (a) Black lines are pairs in H, solid lines are pairs in G, and dashed lines are pairs in G^c. The pair (a,b) in the cyan color is the pair to be labeled or deduced. (b) If $(a,b) \in G$, $\{(a,b),(e,f),(a,f),(e,b)\} \subseteq N_{(a,b)}$. (c) If $(a,b) \in G$, $\{(h,e),(h,a)\} \subseteq T_{(a,b)}$ and $\{(b,g),(f,g)\} \subseteq S_{(a,b)}$. (d) If $(a,b) \in G^c$, $\{(a,b),(a,d),(c,b),(c,d)\} \subseteq N'_{(a,b)}$. Likewise, if $\exists(x,y) \in G$, s.t. $(a,b) \in S_{(x,y)} \cup T_{(x,y)}$, $\{(a,b),(a,d),(c,b)\} \subseteq O_{(x,y)}$.

then moved to $\mathcal{D}_l$. Typically instances are queried in a prioritized way such that one can obtain good classifiers trained with a substantially smaller set D_l. We focus on the pool-based sampling setting where queries are selected in serial, i.e., one at a time.

3.1 Query Strategies

The key component of active learning is the design of an effective criterion for selecting the most "valuable" instance to query, which is often referred to as *query strategy*. We use s^* to refer to the selected instance by the strategy. In general, different strategies follow a greedy framework:

$$s^* = \underset{s \in D_u}{\operatorname{argmax}} \min_{y \in \{-1,1\}} f(s; y, D_l), \tag{1}$$

where $f(s; y, \mathcal{D}_l) \in \mathbb{R}$ is a scoring function to measure the risks of choosing y as the label for $\mathbf{x}_s \in D_u$ given an existing labeled set D_l.

We investigate two commonly used query strategies: uncertainty sampling [6] and query-by-committee [14]. We show that under the binary classification setting, they can all be reformulated as Eq. (1).

Uncertainty Sampling selects the instance which it is least certain how to label. We choose to study one popular uncertainty-based sampling variant, the *least confident*. Subject to Eq. (1), the resulting approach is to let

$$f(s; y, \mathcal{D}_l) = 1 - P_{\Delta(\mathcal{D}_l)}(y_s = y | \mathbf{x}_s), \tag{2}$$

where $P_{\Delta(\mathcal{D}_l)}(y_s = y|\mathbf{x}_s)$ is a conditional probability which is estimated from a probabilistic classification model Δ trained on $\{(\mathbf{x}_i, y_i) \mid \forall i \in \mathcal{D}_l\}$.

Query-By-Committee maintains a committee of models trained on labeled data, $\mathcal{C}(\mathcal{D}_l) = \{g^{(1)}, ..., g^{(C)}\}$. It aims to reduce the size of version space. Specifically, it selects the unlabeled instance about which committee members disagree the most based on their predictions. Subject to Eq. (1), the resulting approach is to let

$$f(s; y, \mathcal{D}_l) = \sum\nolimits_{k=1}^{C} \mathbf{1}[y \neq g^{(k)}(\mathbf{x}_s)], \tag{3}$$

where $g^{(k)}(\mathbf{x}_s) \in \{-1, 1\}$ is the predicted label of $\mathbf{x}_s$ using the classifier $g^{(k)}$.

Our paper will start from generalizing Eq. (1) and show that it is possible to extend the two popular query strategies for considering relational data as a strict order.

4. ACTIVE LEARNING OF A STRICT ORDER

Given G a strict order of V, consider a set of data $\mathcal{D} \subseteq V \times V$, where $(a,a) \notin \mathcal{D}, \forall a \in V$. Similar to the pool-based active learning, one needs to maintain a labeled set D_l and an unlabeled set D_u. We require that $\mathcal{D} \subseteq \mathcal{D}_l \cup \mathcal{D}_u$and $\mathcal{D}_l \cap \mathcal{D}_u = \varnothing$. Given a feature extractor $\mathcal{F} : V \times V \mapsto \mathbb{R}^d$, we can build a vector dataset $\{\mathbf{x}_{(a,b)} = \mathcal{F}(a,b) \in \mathbb{R}^d \mid (a,b) \in \mathcal{D}\}$. Let $y_{(a,b)} = -1 + 2 \cdot \mathbf{1}[(a,b) \in G] \in \{-1, 1\}$ be the ground-truth label for each $(a,b) \in V \times V$. Active learning aims to query Q a subset from $\mathcal{D}$ under limited budget and construct a label set $\mathcal{D}_l$ from Q, in order to train a good classifier h on $\mathcal{D}_l \cap \mathcal{D}$ such that it predicts accurately whether or not an unlabeled pair $(a,b) \in G$ by $h(\mathcal{F}(a,b)) \in \{-1, 1\}$.

Active learning of strict orders differs from the traditional active learning in two unique aspects: (i) By querying the label of a single unlabeled instance, one may obtain a set of labeled examples, with the help of strict orders' properties; (ii) The relational information of strict orders could also be utilized by query strategies. We will present our efforts towards incorporating the above two aspects into active learning of a strict order.

4.1 Basic Relational Reasoning in Active Learning

A basic extension from standard active learning to one under the strict order setting is to apply relational reasoning when both updating $\mathcal{D}_l$ and predicting labels. Algorithm 1 shows the pseudocode for the pool-based active learning of a strict order. When updating $\mathcal{D}_l$ with a new instance $(a,b) \in \mathcal{D}_u$ whose label $y_{(a,b)}$ is acquired from querying, one first calculates $\overline{\mathcal{D}'_l}$, i.e., the closure of $\mathcal{D}_l \cup \{(a,b)\}$, using Theorem 1, and then sets $\mathcal{D}_l := \overline{\mathcal{D}'_l}$ and $\mathcal{D}_u := D \backslash \overline{\mathcal{D}'_l}$ respectively. Therefore, it is possible to augment the labeled set $\mathcal{D}_l$ with more than one pair at each stage even though only a single instance is queried. Furthermore, the following corollary shows that given a fixed set of samples to be queried, their querying order does not affect the final labeled set D_l constructed.

COROLLARY 1.1. *Given a list of pairs Q of size m whose elements are from $V \times V$, let $i_1, \ldots, i_m$ and $j_1, \ldots, j_m$ be two different permutations of $1, \ldots, m$. Let $I_0 = \varnothing$ and $J_0 = \varnothing$, and $I_k = I_{k-1} \cup \{q_{i_k}\}$, $J_k = J_{k-1} \cup \{q_{j_k}\}$ for $k = 1, \ldots, m$, where $\bar{\cdot}$ is defined as the closure set under G. We have $I_m = J_m$, which is the closure of $\{q_i \in V \times V \mid i = 1, \ldots, m\}$.*

Corollary 1.1 is a straightforward result from the uniqueness of closure, which is also verified by our experiments. The la-

Algorithm 1 Pseudocode for pool-based active learning of a strict order.

Input:
$\mathcal{D} \subseteq V \times V$ % a data set
Initialize:
$\mathcal{D}_l \leftarrow \{(a_{s_1}, b_{s_1}), (a_{s_2}, b_{s_2}), ..., (a_{s_k}, b_{s_k})\}$ % initial labeled set with k seeds
$\mathcal{D}_l \leftarrow \overline{\mathcal{D}_l}$ % initial closure
$\mathcal{D}_u \leftarrow \mathcal{D} \backslash \mathcal{D}_l$ % initial unlabeled set
while $\mathcal{D}_u \neq \emptyset$ **do**
Select (a^*, b^*) from $\mathcal{D}_u$ % according to a query strategy
Query the label $y_{(a^*,b^*)}$ for the selected instance (a^*, b^*)
$\mathcal{D}_l \leftarrow \overline{\mathcal{D}_l \cup \{(a^*, b^*)\}}$
$\mathcal{D}_u \leftarrow \mathcal{D} \backslash \mathcal{D}_l$
end while

beled set $\mathcal{D}_l$ contains two kinds of pairs based on where their labels come from: The first kind of labels comes directly from queries, and the second kind comes from the relational reasoning as explained by Theorem 1. Such an approach has a clear advantage over standard active learning at the same budget of queries, because labels of part of the test pairs can be inferred deterministically and as a result there will be more labeled data for supervised training. In our setup of active learning, we train classifiers on $\mathcal{D} \cap \mathcal{D}_l$ and use them for predicting the labels of remaining pairs that are not in $\mathcal{D}_l$.

4.2 Query Strategies with Relational Reasoning

The relational active learning framework as explained in the previous section however does not consider incorporating relational reasoning in its query strategy. We further develop a systematic approach on how to achieve this.

We start from the following formulation: at each stage, one chooses a pair (a^*, b^*) to query based on

$$(a^*, b^*) = \operatorname*{argmax}_{(a,b)\in\mathcal{D}_u} \min_{y\in\{-1,1\}} F(\mathcal{S}(y_{(a,b)} = y), \mathcal{D}_l), \quad (4)$$

$$\mathcal{S}(y_{(a,b)} = y) = (\overline{\mathcal{D}_l \cup \{(a,b)\}} \backslash \mathcal{D}_l) \cap \mathcal{D}. \quad (5)$$

Again, F is the scoring function. $\mathcal{S}(y_{(a,b)} = y)$ is the set of pairs in $\mathcal{D}$ whose labels, originally unknown ($\notin \mathcal{D}_l$), can now be inferred by assuming $y_{(a,b)} = y$ using Theorem 1. For each $(u, v) \in \mathcal{S}(y_{(a,b)} = y)$, its inferred label is denoted as $\hat{y}_{(u,v)}$ in the sequel. One can see that this formulation is a generalization of Eq. (1). We now proceed to develop extensions for the two query strategies to model the dependencies between pairs imposed by the rule of a strict order. Following the same notations as previously described with the only difference that the numbering index is replaced by the pairwise index, we propose two query strategies tailored to strict orders.

Uncertainty Sampling with Reasoning. With relational reasoning, one not only can reduce the uncertainty of the queried pair (a, b) but also may reduce that of other pairs deduced by assuming $y_{(a,b)}$=y. The modified scoring function reads:

$$F(\mathcal{S}(y_{(a,b)} = y), \mathcal{D}_l) = \sum_{(u,v)\in\mathcal{S}(y_{(a,b)}=y)} 1 - P_{\Delta(\mathcal{D}_l\cap\mathcal{D})}(y_{(u,v)} = \hat{y}_{(u,v)} | \mathbf{x}_{(u,v)}). \quad (6)$$

Query-by-Committee with Reasoning. Likewise, one also has the extension for QBC, where $\{g^{(k)}\}_{k=1}^{C}$ is a committee of classifiers trained on bagging samples of $\mathcal{D}_l \cap \mathcal{D}$,

$$F(\mathcal{S}(y_{(a,b)} = y), \mathcal{D}_l) = \sum_{(u,v)\in\mathcal{S}(y_{(a,b)}=y)} \sum_{k=1}^{C} \mathbf{1}(\hat{y}_{(u,v)} \neq g^{(k)}(\mathbf{x}_{(u,v)})). \quad (7)$$

5. EXPERIMENTS

For evaluation, we apply the proposed active learning algorithms to *concept prerequisite learning problem* [8]. Given a pair of concepts (A, B), we predict whether or not A is a prerequisite of B, which is a binary classification problem. Here, cases where B is a prerequisite of A and where no prerequisite relation exists are both considered negative.

5.1 Dataset

We use the Wiki concept map dataset from [17] which is collected from textbooks on different educational domains. Each concept corresponds to an English Wiki article. For each domain, the dataset consists of prerequisite pairs in the concept map. Table 1 summarizes the statistics of the our final processed dataset.

Table 1: Dataset statistics.

Domain	# Concepts	# Pairs	# Prerequisites
Data Mining	120	826	292
Geometry	89	1681	524
Physics	153	1962	487
Precalculus	224	2060	699

5.2 Features

For each concept pair (A, B), we calculate two types of features following the popular practice of information retrieval and natural language processing: graph-based features and text-based features. Please refer to Table 2 for detailed description. Note we trained a topic model [1] on the Wiki corpus. We also trained a Word2Vec [12] model on the same corpus with each concept treated as an individual token.

5.3 Experiment Settings

We follow the typical evaluation protocol of pool-based active learning. We first randomly split a dataset into a training set $\mathcal{D}$ and a test set $\mathcal{D}_{test}$ with a ratio of 2:1. Then we randomly select 20 samples from the training set as the initial query set Q and compute its closure $\mathcal{D}_l$. Meanwhile, we set $\mathcal{D}_u = \mathcal{D} \backslash \mathcal{D}_l$. In each iteration, we pick an unlabeled instance from $\mathcal{D}_u$ to query for its label, update the label set $\mathcal{D}_l$, and re-train a classification model on the updated $\mathcal{D}_l \cap \mathcal{D}$. The re-trained classification model is then evaluated on $\mathcal{D}_{test}$. In all experiments, we use a random forests classifier [2] with 200 trees as the classification model. We use Area under the ROC curve (AUC) as the evaluation metric. Taking into account the effects of randomness subject to different initializations, we continue the above experimental process for each method repeatedly with 300 preselected distinct random seeds. Their average scores and

Table 2: Feature description. Top: graph-based features. Bottom: text-based features.

Feature	Description
In/Out Degree	The in/out degree of A/B.
Common Neighbors	# common neighbors of A and B.
# Links	# times A/B links to B/A.
Link Proportion	The proportion of pages that link to A/B also link to B/A.
NGD	The Normalized Google Distance between A and B [18].
PMI	The Pointwise Mutual Information relatedness between the incoming links of A and B.
RefD	A metric to measure how differently A and B's related concepts refer to each other [8].
HITS	The difference between A and B's hub/authority scores. [5]
1st Sent	Whether A/B is in the first sentence of B/A.
In Title	Whether A appears in B's title.
Title Jaccard	The Jaccard similarity between A and B's titles.
Length	# words of A/B's content.
Mention	# times A/B are mentioned in the content of B/A.
NP	# noun phrases in A/B's content; # common noun phrases.
Tf-idf Sim	The cosine similarity between Tf-idf vectors for A and B's first paragraphs.
Word2Vec Sim	The cosine similarity between vectors of A and B trained by Word2Vec.
LDA Entropy	The Shannon entropy of the LDA vector of A/B.
LDA Cross Entropy	The cross entropy between the LDA vector of A/B and B/A.

Table 3: Summary of compared query strategies.

Method	Use reasoning when updating $\mathcal{D}_l$	Use reasoning to select the instance to query	Use learning to select the instance to query
Random	✗	✗	✗
LC, QBC	✗	✗	✓
Random-R	✓	✗	✗
LC-R, QBC-R	✓	✗	✓
CNT	✓	✓	✗
LC-R+, QBC-R+	✓	✓	✓

confidence intervals ($\alpha = 0.05$) are reported. We compare four query strategies: (i) **Random**: randomly select an instance to query; (ii) **LC**: least confident sampling, a widely used uncertainty sampling variant. We use logistic regression to estimate posterior probabilities; (iii) **QBC**: query-by-committee algorithm. We apply query-by-bagging [11] and use a committee of three decision trees; (iv) **CNT**: a simple baseline query strategy designed to greedily select an instance whose label can potentially infer the most number of unlabeled instances. Following the previous notations, the scoring function for CNT is $F(\mathcal{S}(y_{(a,b)} = y), \mathcal{D}_l) = |S(y_{(a,b)} = y)|$, which is solely based on logical reasoning.

For experiments, we test each query strategy under three settings: (i) Traditional active learning where no relational information is considered. Query strategies under this setting are denoted as Random, LC, and QBC. (ii) Relational active learning where relation reasoning is applied to updating $\mathcal{D}_l$ and predicting labels of $\mathcal{D}_{test}$. Query strategies under this setting are denoted as Random-R, LC-R, and QBC-R. (iii) Besides being applied to updating $\mathcal{D}_l$, relational reasoning is also incorporated in the query strategies. Query strategies under this setting are the baseline method CNT and our proposed extensions of LC and QBC for strict partial orders, denoted as LC-R+ and QBC-R+, respectively. Table 3 summarizes the query strategies studied in the experiments.

5.4 Experiment Results

Figure 2 shows the AUC results of different query strategies. For each case, we present the average values and 95% C.I. of repeated 300 trials with different train/test splits. In addition, Figure 3 compares the relations between the number of queries and the number of labeled instances across different query strategies. Note that in the relational active learning setting querying a single unlabeled instance will result in one or more labeled instances. According to Figure 2 and Figure 3, we have the following observations: First, by comparing query strategies under the settings (ii) and (iii) with setting (i), we observe that incorporating relational reasoning into active learning substantially improves the AUC performance of each query strategy. In addition, we find the query order, which is supposed to be different for each strategy, does not affect $\mathcal{D}_l$ at the end when $\mathcal{D} \subseteq \mathcal{D}_l$. Thus, it partly verifies Corollary 1.1. Second, our proposed LC-R+ and QBC-R+ significantly outperform other compared query strategies. Specifically, when comparing them with LC-R and QBC-R, we see that incorporating relational reasoning into directing the queries helps to train a better classifier. Figure 3 shows that LC-R+ and QBC-R+ lead to more labeled instances when using the same amount of queries than that of LC-R and QBC-R. This partly contributes to the performance gain. Third, LC-R+ and QBC-R+ are more effective at both collecting a larger labeled set and training better classifiers than the CNT baseline. In addition, by comparing CNT with LC-R, QBC-R, and Random-R, we observe that a larger size of the labeled set does not always lead to a better performance. Such observations demonstrate the necessity of combining deterministic relational reasoning and probabilistic machine learning in designing query strategies.

In addition to effectiveness, we also conduct empirical studies on the runtime of the reasoning module and include the results in the supplemental material.

6. CONCLUSION

We propose an active learning framework tailored to relational data in the form of strict partial orders. An efficient reasoning module is proposed to extend two commonly used query strategies – uncertainty sampling and query by committee. Experiments on concept prerequisite learning show that incorporating relational reasoning in both selecting valuable examples to label and expanding the training set significantly improves standard active learning approaches. Future work could be to explore the following: (i) apply the reasoning module to extend other query strategies; (ii) active learning of strict partial orders from a noisy oracle.

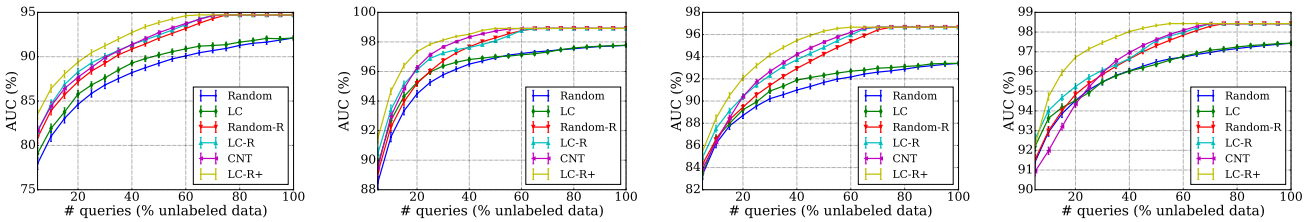
7. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

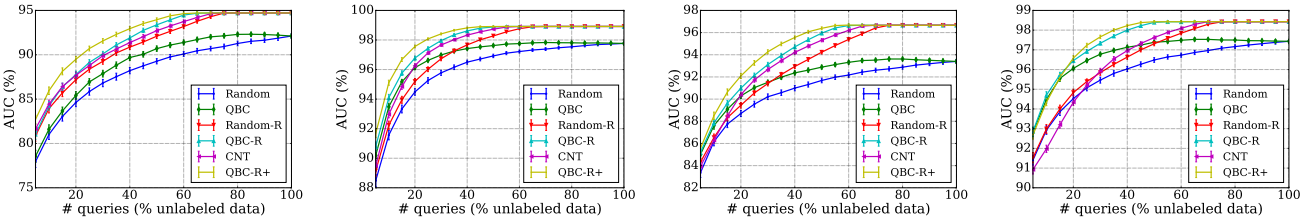
[2] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4(1):129–145, 1996.

[4] P. Donmez and J. G. Carbonell. Optimizing estimated loss reduction for active sampling in rank learning. In *Proc. ICML*, pages 248–255. ACM, 2008.

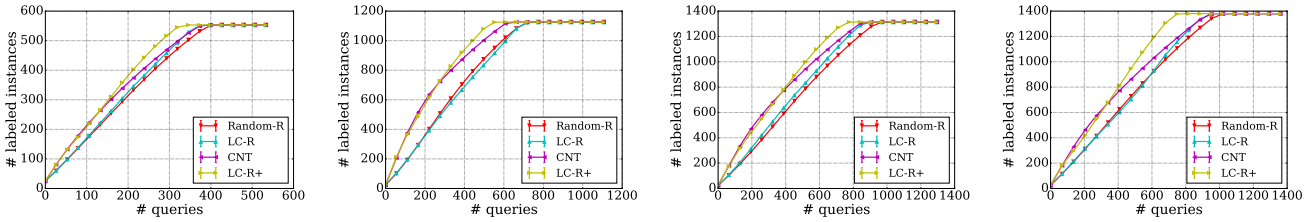


(a) Data Mining - LC (b) Geometry - LC (c) Physics - LC (d) Precalculus - LC

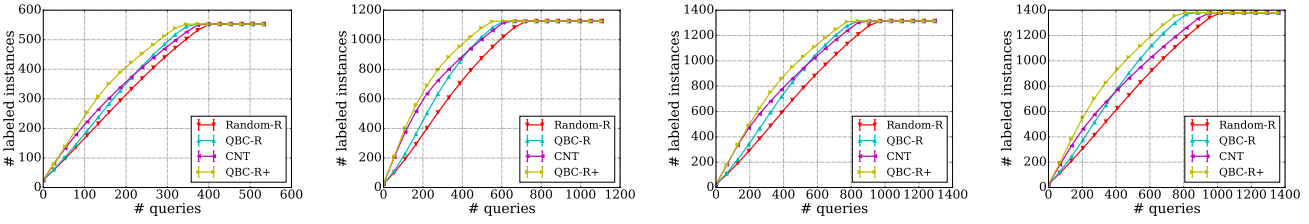


(e) Data Mining - QBC (f) Geometry -QBC (g) Physics - QBC (h) Precalculus - QBC

Figure 2: Comparison of different query strategies' AUC scores for concept prerequisite learning.



(a) Data Mining - LC (b) Geometry - LC (c) Physics - LC (d) Precalculus - LC



(e) Data Mining - QBC (f) Geometry - QBC (g) Physics - QBC (h) Precalculus - QBC

Figure 3: Comparison of relations between the number of queries and the number of labeled instances when using different query strategies.

[5] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[6] D. D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proc. ICML*, pages 148–156, 1994.

[7] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proc. SIGIR*, pages 3–12, 1994.

[8] C. Liang, Z. Wu, W. Huang, and C. L. Giles. Measuring prerequisite relations among concepts. In *Proc. EMNLP*, pages 1668–1674, 2015.

[9] C. Liang, J. Ye, S. Wang, B. Pursel, and C. L. Giles. Investigating active learning for concept prerequisite learning. In *Proc. EAAI*, 2018.

[10] T.-Y. Liu. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

[11] N. A. H. Mamitsuka. Query learning strategies using boosting and bagging. In *Proc. ICML*, volume 1. Morgan Kaufmann Pub, 1998.

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proc. NIPS*, pages 3111–3119, 2013.

[13] S. Rendle and L. Schmidt-Thieme. Active learning of equivalence relations by minimizing the expected loss using constraint inference. In *Proc. ICDM*, pages 1001–1006. IEEE, 2008.

[14] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proc. COLT*, pages 287–294. ACM, 1992.

[15] P. P. Talukdar and W. W. Cohen. Crowdsourced comprehension: predicting prerequisite structure in Wikipedia. In *Proc. BEA*, pages 307–315. ACL, 2012.

[16] A. Vuong, T. Nixon, and B. Towle. A method for finding prerequisites within a curriculum. In *Proc. EDM*, pages 211–216, 2011.

[17] S. Wang, A. Ororbia, Z. Wu, K. Williams, C. Liang, B. Pursel, and C. L. Giles. Using prerequisites to extract concept maps from textbooks. In *Proc. CIKM*, pages 317–326. ACM, 2016.

[18] I. Witten and D. Milne. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, pages 25–30, 2008.