Investigating Active Learning for Concept Prerequisite Learning

Chen Liang **Jianbo Ye** **Shuting Wang** **Bart Pursel** **C. Lee Giles**
Pennsylvania State University
University Park, PA 16801
{cul226, jxy198, bkp10, giles}@ist.psu.edu sxw327@cse.psu.edu

Abstract

Concept prerequisite learning focuses on machine learning methods for measuring the prerequisite relation among concepts. With the importance of prerequisites for education, it has recently become a promising research direction. A major obstacle to extracting prerequisites at scale is the lack of large scale labels which will enable effective data driven solutions. We investigate the applicability of active learning to concept prerequisite learning. We propose a novel set of features tailored for prerequisite classification and compare the effectiveness of four widely used query strategies. Experimental results for domains including data mining, geometry, physics, and precalculus show that active learning can be used to reduce the amount of training data required. Given the proposed features, the query-by-committee strategy outperforms other compared query strategies.

Introduction

A *prerequisite* relation describes a fundamental directed relation among concepts in knowledge structures. Following the learning order that is consistent with the underlying prerequisite relations is also crucial to successful teaching and learning processes. For the example shown in Figure 1, learning the concept “Hidden Markov model” requires first understanding its prerequisites such as “posterior probability” and “maximum likelihood”. Obtaining prerequisite relations is crucial for a variety of other educational applications such as curriculum planning (Agrawal, Golshan, and Papalexakis 2015) and intelligent tutoring systems (Aleven and Koedinger 2002). It can be especially useful for online learning at scale where students are faced with a large amount of educational resources.

This paper focuses on the *concept prerequisite learning problem* (Talukdar and Cohen 2012; Liang et al. 2015), where the goal is to predict whether a concept A is a prerequisite of a concept B given the pair (A, B). Although there has been research on learning prerequisites (Vuong, Nixon, and Towle 2011; Talukdar and Cohen 2012; Liang et al. 2015; Wang et al. 2016; Scheines, Silver, and Goldin 2014; Liu et al. 2016; Pan et al. 2017), the lack of large scale prerequisite labels remains a major obstacle for effective machine learning-based solutions.

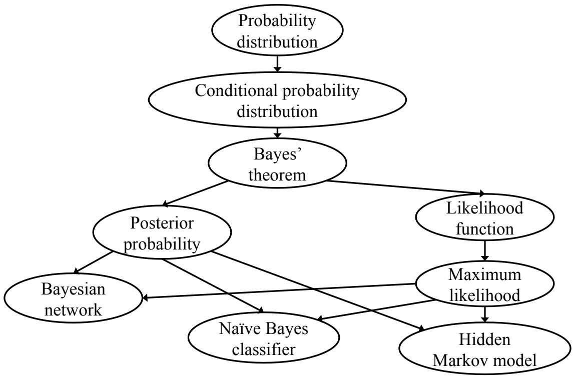


Figure 1: Concept prerequisite relations. “$A \rightarrow B$” represents that the concept A is a prerequisite of the concept B.

A possible solution for learning a good classifier given limited labeled instances is active learning (Angluin 1988; Cohn, Ghahramani, and Jordan 1996; Settles 2010), since it is designed to learn classifiers with significantly fewer labels by actively directing the query to the most “valuable” examples. As such, active learning methods could could also be applied to solving the current challenges of concept prerequisite learning.

The goal of this work is to make the first attempt of applying active learning to the concept prerequisite learning problem. We investigate three families of query selection strategies by comparing their effectiveness on reducing the amount of training data. The first are informativeness-based methods such as uncertainty sampling (Lewis and Catlett 1994) and query-by-committee (Seung, Opper, and Sompolinsky 1992). The second are methods which take both informativeness and representativeness into account. The third are diversity-based strategies which aim to cover the feature space as broadly as possible. For classification, we propose a novel set of features for representing concept pairs and choose from four widely used classifiers the most suitable one for conducting active learning experiments. Our experiment results show a clear win for query-by-committee over other compared query strategies and show that active learning can be used to reduce the amount of training data required for concept prerequisite learning.

Related Work

Regardless of being a relatively new research area, data-driven methods for learning concept prerequisite relations have been explored in multiple works. Established methods in educational data mining have been devoted to analyzing student assessment data which records the performance of students on different items (Vuong, Nixon, and Towle 2011; Scheines, Silver, and Goldin 2014; Chen, Wuillemin, and Labat 2015; Chen, González-Brenes, and Tian 2016). Such methods require that the association between test items and handcrafted knowledge components is set beforehand and are not applicable for processing a large concept set. Gordon et al. (2016) proposed an information-theoretic metric to capture concept dependencies in a scientific corpus. Their method relies on topic modeling techniques and requires human annotations of latent topics to make the result interpretable. Wikipedia has been exploited to find prerequisite relations among universally shared concepts. (Talukdar and Cohen 2012; Liang et al. 2015; Wang et al. 2016; Agrawal, Golshan, and Papalexakis 2015), using both the Wikipedia article content and their linkage structures. Besides information in Wikipedia, Pan et al. (2017) propose to include other features such as video references and sentence references for learning prerequisite relations among concepts in MOOCs. In addition, course prerequisites have also been used for learning concept prerequisite relations (Yang et al. 2015; Liang et al. 2017). To our knowledge, active learning has not been applied to the concept prerequisite learning problem.

Pool-based Active Learning

Pool-based sampling (Lewis and Gale 1994) 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 ground truth 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 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. Algorithm 1 presents the typical setting of serial pool-based active learning.

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.

Algorithm 1 Pseudocode for pool-based active learning.

```
Input:
  $\mathcal{D} \leftarrow \{1, 2, ..., n\}$   % a data set of $n$ instances
Initialize:
  $\mathcal{D}_l \leftarrow \{s_1, s_2, ..., s_k\}$   % initial labeled set with $k$ seeds
  $\mathcal{D}_u \leftarrow \mathcal{D} \backslash \mathcal{D}_l$   % initial unlabeled set
while $\mathcal{D}_u \neq \emptyset$ do
    Select $s^*$ from $\mathcal{D}_u$   % according to a query strategy
    Query the label $y_{s^*}$ for the selected instance $s^*$
    $\mathcal{D}_l \leftarrow \mathcal{D}_l \cup \{s^*\}$
    $\mathcal{D}_u \leftarrow \mathcal{D}_u \backslash \{s^*\}$
end while
```

We investigate four commonly used query strategies: uncertainty sampling (Lewis and Catlett 1994), query-by-committee (Seung, Opper, and Sompolinsky 1992), QUIRE (Huang, Jin, and Zhou 2014), and diversity sampling. These strategies are designed based on different assumptions: The first two selection strategies are based on the informativeness of the instance estimated by classifiers; QUIRE is based on the combination of informativeness and representativeness; Diversity sampling is based on the diversity in the feature space. Although being different, 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

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

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

QUIRE aims to measure and combine the two types of query selection criteria, informativeness and representativeness, using a comprehensive max-margin framework. Subject to Eq. (1), the resulting approach is to let

$$\begin{aligned} f(s; y, \mathcal{D}_l) = (L_{u,l}\mathbf{y}_l + L_{u,s}y)^T L_{u,u}^{-1} (L_{u,l}\mathbf{y}_l + L_{u,s}y) \\ - 2yL_{s,l}\mathbf{y}_l - L_{s,s}, \end{aligned} \tag{4}$$

where $L = (K + \lambda I)^{-1}$ and K is the kernel matrix of size $n \times n$ and $f(s; y, \mathcal{D}_l)$ is equal to the *negative* margin if $y_s = y$ up to a constant. Due to limited space, please refer to (Huang, Jin, and Zhou 2014) (pp. 1938–1940) for their detailed notations.

Diversity Sampling aims to select instances that cover as much of the feature space as possible. It selects the unlabeled instance with the lowest average cosine similarity between the instance's feature vector and those of the instances in the current training labeled dataset. Subject to Eq. (1), the resulting approach is to let

$$f(s; y, \mathcal{D}_l) = \sum_{i \in \mathcal{D}_l} 1 - \cos(\mathbf{x}_s, \mathbf{x}_i) \tag{5}$$

where $\cos(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{|\mathbf{x}_i||\mathbf{x}_j|}$ is the cosine similarity function. Note label y is not considered in this method.

Experimental Design

For experiments, we apply the previously mentioned active learning algorithms to *concept prerequisite learning problem* (Liang et al. 2015). 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.

Dataset

We use the Wiki concept map dataset (Wang et al. 2016) which is collected from textbooks on four different educational domains. For each domain, the dataset consists of prerequisite pairs in the concept map. In the preprocessing stage, we validate whether each of the prerequisite relations in the dataset satisfies the required properties of a strict partial order (i.e., transitivity and irreflexivity) and ask domain experts to manually correct their labels if needed. We also expand the dataset by using the irreflexive and transitive properties: (i) add (B, A) as a negative sample if (A, B) is a positive sample; (ii) add (A, C) as a positive sample if both (A, B) and (B, C) are positive samples. Table 1 summarizes the statistics of the our final processed dataset.

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

Table 1: Dataset statistics.

Feature Description

For each concept pair (A, B), we calculate two types of features from information retrieval and natural language processing: graph-based and text-based features. Note that for all features, we use a Wikipedia dump of Oct. 2016.

Graph-based Features (GF) The first type of feature is designed to utilize the link structure of Wikipedia. For convenience, we use the following notations: $In(A)$ is the set of concepts that link to A; $Out(A)$ is the set of concepts which A links to; $\mathcal{C} = \{c_1, ..., c_N\}$ is the concept space, i.e. all concepts in Wikipedia. Specifically, different types of graph-based features are:

- In/Out Degree. (GF #1-#4) The in/out degree of A/B.
- Common Neighbors. (GF #5) The number of common neighbors of A and B, i.e. $|Out(A) \cap Out(B)|$.
- # Links. (GF #6-#7) The number of times A/B links to B/A. The link structure within Wikipedia can be used as a proxy for prerequisite relations. The intuition is that a concept is usually linked to its prerequisites.
- Link Proportion. (GF #8-#9) The proportion of pages that link to A/B also link to B/A, i.e. $\frac{|In(A) \cap In(B)|}{|In(A)|}$ and $\frac{|In(A) \cap In(B)|}{|In(B)|}$.
- NGD. (GF #10) The Normalized Google Distance (Witten and Milne 2008) between A and B. Specifically,

  $$NGD(A, B) = \frac{\max(\log |In(A)|, \log |In(B)|) - \log |In(A) \cap In(B)|}{\log N - \min(\log |In(A)|, \log |In(B)|)}$$

- PMI. (GF #11) The Pointwise Mutual Information relatedness between the incoming links of A and B. (Ratinov et al. 2011)

  $$PMI(A, B) = \log \frac{N \cdot |In(A) \cap In(B)|}{|In(A)| \cdot |In(B)|}$$

- RefD. (GF #12) A metric to measure how differently A and B's related concepts link to each other. Proposed by (Liang et al. 2015), RefD has been used as a proxy to measure concept prerequisite relations.

  $$RefD(A, B) = \frac{\sum_{i=1}^{N} r(c_i, B) \cdot w(c_i, A)}{\sum_{i=1}^{N} w(c_i, A)} - \frac{\sum_{i=1}^{N} r(c_i, A) \cdot w(c_i, B)}{\sum_{i=1}^{N} w(c_i, B)}$$

  where $w(c_i, A)$ weights the importance of c_i to A; and $r(c_i, A)$ is an indicator showing whether c_i links to A.
- PageRank. (GF #13) The difference between A and B's PageRank scores (Page et al. 1999). The PageRank score, based on the link analysis, can be used to estimate the importance of concepts.
- HITS. (GF #14-#15) The difference between A and B's hub scores and the difference between their authority scores (Kleinberg 1999). Similar to PageRank, authority and hub scores are used as proxies for concept importance.

Text-based Features (TF) The second type of feature is designed to utilize textual content in the Wikipedia page. Note we have trained a topic model (Blei, Ng, and Jordan 2003) (#topics=300) on the Wiki corpus. We have also trained a word2vec (Mikolov et al. 2013) (size=300) model on the same corpus with each concept treated as an individual token. Specifically, different types of text-based features are:

- 1st Sent. (TF #1-#2) Whether A/B appears in the first sentence of B/A. The first sentence of a Wikipedia article is usually the definition of the concept and the concepts mentioned in the sentence are more likely to be a prerequisite.
- In Title. (TF #3) Whether A appears in B's title. For example, "machine learning" is contained in "Supervised machine learning".
- Title Jaccard. (TF #4) The Jaccard similarity between A and B's titles.
- Length. (TF #5-#6) The number of words of A/B's content. This might serve as a proxy for complexity level and popularity of the concept.
- Mention. (TF #7-#8) The number of times A/B are mentioned in the content of B/A. The intuition is that the important prerequisites of a concept might be mentioned many times in its content.
- NP. (TF #9-#11) The number of noun phrases in A/B's content; The number of common noun phrases. If the concept is very general, its content tends to have more noun phrases.
- Tf-idf Sim. (TF #12) The cosine similarity between Tf-idf vectors for A and B's first paragraphs.
- Word2vec Sim. (TF #13) The cosine similarity between vectors of A and B trained by word2vec. Both word2vec and tf-idf similarities are measures for semantic relatedness, which is needed because usually two concepts with prerequisite relation are semantically related.
- LDA Entropy. (TF #14-#15) The Shannon entropy of the LDA vector of A/B.

  $$H(A) = -\sum_{i}^{T} p_{Ai} \log p_{Ai}$$

  where $\mathbf{p}_A$ is A's LDA vector, i.e., the distribution over T topics. More advanced concepts usually focus on fewer topics, thus leading to a lower LDA entropy.
- LDA Cross Entropy. (TF #16-#17) The cross entropy between the LDA vector of A/B and B/A. Gordon et al. (2016) propose to use this feature to capture concept dependencies in a scientific corpus.

  $$H(A;B) = H(A) + D_{KL}(A||B)$$

  where $H(A)$ is the entropy of A's LDA vector, and $D_{KL}(A||B)$ is the Kullback-Leibler divergence between A and B's LDA vectors.

Experimental Results

Evaluation of Classification

Before investigating the performance of active learning, we first evaluate concept prerequisite learning under the traditional binary classification setting. In our experiments, we employ four widely used binary classifiers: Naïve Bayes (NB), Logistic Regression (LR), Support Vector Machine (SVM) (Cortes and Vapnik 1995), and Random Forest (RF) (Breiman 2001). Specifically, we set $C = 1.0$ for LR, use a linear kernel for SVM, and use 200 trees for RF. For each dataset, we apply 5-fold cross validation and report the average precision (P), recall (R), F1-score ($F1$) and Area under the ROC curve (AUC).

Classifier	Metric	Data Mining	Geometry	Physics	Precalulus
NB	P	71.5	84.4	54.3	85.7
	R	28.5	44.3	71.9	66.9
	$F1$	37.8	58.1	61.6	75.0
	AUC	81.4	87.1	85.5	93.2
LR	P	65.8	71.3	58.0	81.7
	R	**77.4**	81.3	**78.8**	**88.4**
	$F1$	71.1	75.8	66.8	84.8
	AUC	85.9	91.6	89.2	95.4
SVM	P	73.7	82.8	77.9	86.7
	R	64.7	69.9	50.3	81.4
	$F1$	68.6	75.5	61.1	83.9
	AUC	85.0	91.3	88.8	95.1
RF	P	**80.7**	**95.0**	**85.2**	**90.2**
	R	73.3	**84.7**	59.3	87.1
	$F1$	**76.7**	**89.5**	**69.9**	**88.6**
	AUC	**92.2**	**97.8**	**93.9**	**97.5**

Table 2: Results (%) for concept prerequisite relation classification.

As shown in Table 2, the classification results vary by different methods. Overall, Naïve Bayes performs the worst in terms of both $F1$ and AUC, which is due to the fact that the strong independence assumption does not hold for our designed feature set. For example, the number of noun phrases might be correlated with the number of words; PageRank and HITS scores are not independent either. As linear classification models, LR and SVM lead to similar $F1$ and AUC while the former has higher recall and the latter has higher precision. Among four methods, Random Forest outperforms other three across all four domains, by 5.6%, 13.7%, 3.1%, and 3.8% respectively w.r.t. $F1$ and 6.3%, 6.1%, 4.7%, and 2.1% w.r.t. AUC. This might be because, compared with a linear combination of features for classification, the procedure of RF (the bagging and random selection of feature set) is more suitable for capturing the relation between the proposed feature set and concept prerequisite relations. We use RF in the following experiments as the classification model.

Feature Analysis

We also conduct a feature analysis in order to gain more insights on the proposed feature set. Table 3 lists top 10 features for each domain. Since Random Forest is used, the feature importance is calculated by "mean decrease impurity". It is defined as the total decrease in node impurity, weighted by the probability of reaching that node, averaged over all trees of the ensemble. From Table 3, we can observe the following: (i) While the ranking of features is different across four domains, there are many common important features such as PageRank, HITS's authority score, RefD, etc; (ii) Among top features, there are more graph-based features than text-based features. This might be because current text-based features are still very simple and more ef-

Data Mining	Geometry	Physics	Precalculus
Authority diff	PageRank diff	PageRank diff	PageRank diff
LDA entropy of A	In degree of A	RefD	Authority diff
PageRank diff	Out degree of A	# mentions of A in B	RefD
In degree of A	RefD	In degree of B	# mentions of A in B
RefD	# mentions of A in B	Authority diff	Out degree of A
LDA entropy of B	LDA entropy of A	Link proportion of B	A in B's 1st sentence
In degree of B	A in B's 1st sentence	Out degree of A	In degree of A
LDA cross entropy (A;B)	Length of A	In degree of A	Hub diff
Link proportion of A	# NPs in A	LDA entropy of A	# NPs in A
LDA cross entropy (B;A)	# mentions of B in A	# NPs in B	# mentions of A in B

Table 3: Top 10 important features for each domain.

fective text features are yet to be explored. Several possible choices include lexico-syntactic patterns (Hearst 1992), structural features (Pan et al. 2017), etc. (iii) Top text-based features are LDA entropy, LDA cross entropy, Mention, and NP. Similarity-based features such as Tf-idf and Word2vec similarities are not as important; (iv) Top graph-based features are PageRank, authority score, RefD, in/out degree, and link proportion. Other graph features such as common neighbors, NGD, and PMI are less important. From observation (iii) and (iv) we find that symmetric pairwise features such as similarity and PMI are not important in current in-domain classification setting. This can be explained by noticing that the motivation of designing these features is to add constraints on the semantic relatedness, which is usually already satisfied in the in-domain setting. We expect such features to be more important in a cross-domain classification setting, where the concept space is much larger and more diverse.

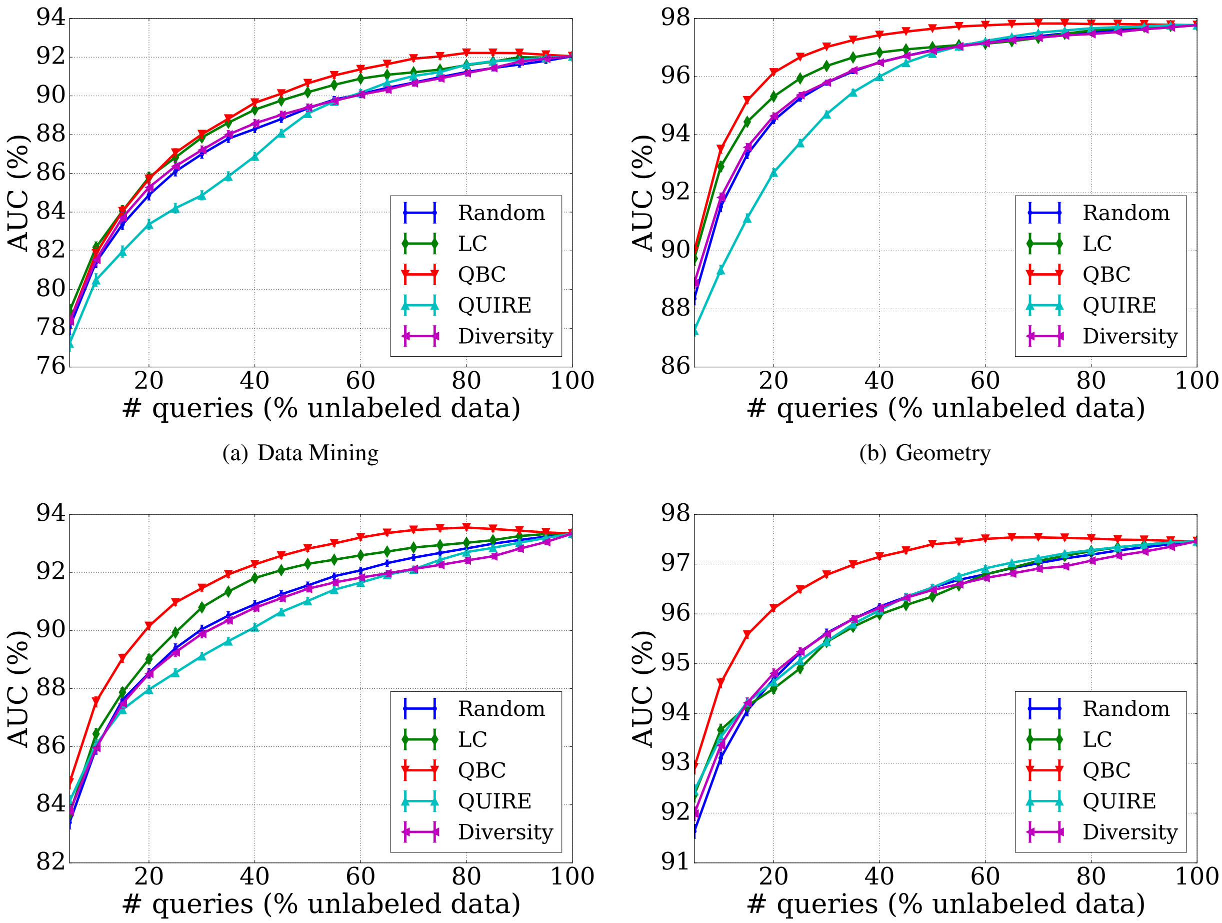
Evaluation of Active Learning

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 (Breiman 2001) 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 confidence intervals ($\alpha = 0.05$) are reported. We compare the following five query strategies, most of which have been introduced in previous sections:

- Random: randomly selecting an instance to query. We choose this as the baseline for comparison.
- LC: least confident sampling, a widely used uncertainty sampling variant. We use a logistic regression model to estimate the posterior probabilities.
- QBC: query-by-committee algorithm. We apply query-by-bagging (Mamitsuka 1998) and use a committee of three decision trees.
- QUIRE: a strategy for querying informative and representative examples. We follow the authors' experimental approach to use an RBF kernel and set the parameter $\lambda = 1$.
- Diversity: a strategy for selecting the unlabeled instance that is as diverse as possible in the feature space of current labeled set.

Results Figure 2 shows the AUC results of different query strategies for concept prerequisite learning. For each case, we present the average values and 95% confidence intervals of repeated 300 trials with different train/test splits. From the figure we have the following observations:

First, comparing results on four domains, we can find different query strategies have relatively consistent learning curves, with the only exception of LC on the Precalculus domain. This is possibly caused by that the number of concept in Precalculus is much larger than other domains and the logistic regression classifier used by LC failed to give an accurate uncertainty estimation. Second, when comparing different strategies with the Random baseline, we find: (i) Informativeness-based methods (least confident sampling and query-by-committee) show substantial improvement over random; QBC is constantly outperforming other query strategies on all domains, which shows the advantage of using ensemble method to estimate uncertainty over the single linear classification model as used by LC. This again suggests that decision tree-based classifiers are more effective given the proposed feature set. (ii) Diversity-sampling is not significantly different from random, which suggests that choosing the instance as diverse as possible in the proposed feature space is not effective. (iii) QUIRE performs worse than random, especially during the early stage of active learning. It is also worth mentioning that QUIRE requires significantly longer time for choosing instances because of calculating the inverse and determinant of large matrices. In addition, for our datasets, by empirically tuning the RBF parameter γ for the best, we still did not find any advantages of QUIRE over LC or QBC. This might be because



(a) Data Mining

(b) Geometry

(c) Physics

(d) Precalculus

Figure 2: Comparison of different query strategies for concept prerequisite classification.

the used RBF kernel, on which QUIRE's performance is critically dependent, does not really suit our provided features.

To sum up, we find that informativeness-based query strategies, especially query-by-committee, is more effective for concept prerequisite learning given the proposed feature set. Different active learning strategies can be used to reduce the amount of training data required to get an expected AUC score for concept prerequisite learning.

Conclusion and Future Work

We made several contributions to concept prerequisite learning. In order to deal with the lack of large scale labels which makes problematic supervised learning for concept prerequisite learning, we investigated the applicability of active learning. Our active learning experiments for comparing different query strategies found that query-by-committee constantly outperforms other methods including uncertainty sampling, QUIRE, and diversity sampling. We proposed a novel set of features for concept pair representation tailored for the concept prerequisite learning problem. The top features identified by the feature importance analysis hopefully will be helpful for other supervised prerequisite learning methods.

Future work could be to design active learning query strategies better tailored to the concept prerequisite learning problem. In the typical setup of active learning, the dependency among labeled or unlabeled instances is not considered. However, since the prerequisite relation is both transitive and irreflexive, then when an unlabeled instance is labeled, there could be other unlabeled instances whose labels can be deduced by applying logical reasoning with the two properties. Query strategies that can take such properties into account will make active learning more effective.

It would be useful to investigate in more detail the semantic representation of concept pairs for prerequisite learning and to design more complex features such as complexity level features, structural features, etc. and see their effect on the classification performance.

Acknowledgements

We gratefully acknowledge partial support from the Pennsylvania State University Center for Online Innovation in Learning.

References

Agrawal, R.; Golshan, B.; and Papalexakis, E. 2015. Data-driven synthesis of study plans. Technical Report TR-2015-003, Data Insights Laboratories.

Aleven, V. A., and Koedinger, K. R. 2002. An effective metacognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor. *Cognitive science* 26(2):147–179.

Angluin, D. 1988. Queries and concept learning. *Machine learning* 2(4):319–342.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent Dirichlet allocation. *JMLR* 3(Jan):993–1022.

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

Chen, Y.; González-Brenes, J. P.; and Tian, J. 2016. Joint discovery of skill prerequisite graphs and student models. In *EDM*, 46–53.

Chen, Y.; Wuillemin, P.; and Labat, J. 2015. Discovering prerequisite structure of skills through probabilistic association rules mining. In *EDM*, 117–124.

Cohn, D. A.; Ghahramani, Z.; and Jordan, M. I. 1996. Active learning with statistical models. *JAIR* 4(1):129–145.

Cortes, C., and Vapnik, V. 1995. Support-vector networks. *Machine learning* 20(3):273–297.

Gordon, J.; Zhu, L.; Galstyan, A.; Natarajan, P.; and Burns, G. 2016. Modeling concept dependencies in a scientific corpus. In *ACL*.

Hearst, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, 539–545. Association for Computational Linguistics.

Huang, S.; Jin, R.; and Zhou, Z. 2014. Active learning by querying informative and representative examples. *IEEE TPAMI* 36(10):1936–1949.

Kleinberg, J. M. 1999. Authoritative sources in a hyperlinked environment. *JACM* 46(5):604–632.

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

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

Liang, C.; Wu, Z.; Huang, W.; and Giles, C. L. 2015. Measuring prerequisite relations among concepts. In *EMNLP*, 1668–1674.

Liang, C.; Ye, J.; Wu, Z.; Pursel, B.; and Giles, C. L. 2017. Recovering concept prerequisite relations from university course dependencies. In *AAAI*, 4786–4791.

Liu, H.; Ma, W.; Yang, Y.; and Carbonell, J. 2016. Learning concept graphs from online educational data. *JAIR* 55:1059–1090.

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

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

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Pan, L.; Li, C.; Li, J.; and Tang, J. 2017. Prerequisite relation learning for concepts in moocs. In *ACL*. ACL.

Ratinov, L.; Roth, D.; Downey, D.; and Anderson, M. 2011. Local and global algorithms for disambiguation to Wikipedia. In *ACL*, 1375–1384. ACL.

Scheines, R.; Silver, E.; and Goldin, I. 2014. Discovering prerequisite relationships among knowledge components. In *EDM*.

Settles, B. 2010. Active learning literature survey. *University of Wisconsin, Madison* 52(55-66):11.

Seung, H. S.; Opper, M.; and Sompolinsky, H. 1992. Query by committee. In *COLT*, 287–294. ACM.

Talukdar, P. P., and Cohen, W. W. 2012. Crowdsourced comprehension: predicting prerequisite structure in Wikipedia. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, 307–315. ACL.

Vuong, A.; Nixon, T.; and Towle, B. 2011. A method for finding prerequisites within a curriculum. In *EDM*.

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

Witten, I., and Milne, D. 2008. 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*, 25–30.

Yang, Y.; Liu, H.; Carbonell, J.; and Ma, W. 2015. Concept graph learning from educational data. In *WSDM*, 159–168.