南京大學

研 究 生 毕 业 论 文

（申 请 博 士 学 位）

论 文 题 目 一阶逻辑领域知识与机器学习的结合研究

作 者 姓 名 戴望州

学科、专业方向 计算机科学与技术

研 究 方 向 机器学习

指 导 教 师 周志华 教授

2018 年 **12** 月 **30** 日

学　　号：**DG1233003**

论文答辩日期：**2019** 年 **2** 月 **26** 日

指 导 教 师：　　（签字）

Research on Integrating First-Order Logical Domain Knowledge with Machine Learning

by

Wang-Zhou Dai

Supervised by

Professor Zhi-Hua Zhou

A dissertation submitted to
the graduate school of Nanjing University
in partial fulfilment of the requirements for the degree of
DOCTOR OF PHILOSOPHY
in
Computer Science and Technology

Department of Computer Science and Technology
Nanjing University

Dec 30, 2018

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：一阶逻辑领域知识与机器学习的结合研究

计算机科学与技术 专业 2012 级博士生姓名：戴望州

指导教师（姓名、职称）：周志华 教授

摘　要

在许多现实的机器学习任务中往往存在着大量的领域知识，且这些知识可以通过一阶逻辑语言表达。本文分别研究了如何使用此类领域知识辅助机器学习，如何利用机器学习进行领域知识精化，以及如何将领域知识与机器学习进行互促结合。针这三个问题，本文提出了四种机器学习方法与框架。具体来说，本文的主要工作如下：

1. **提出一种领域知识增广样本的机器学习方法。**常规的机器学习方法难以对领域中对象间的关系信息进行利用，一阶逻辑语言可以方便地将这种信息表达为领域知识。本文提出的 SUL 学习方法能够将此类领域知识转化为训练样本来提升机器学习性能。简单来说，它能够将领域知识增广为机器学习可直接利用的样本特征。实验表明，相对于未引入领域知识的方法，SUL 学习能够提升机器学习性能。
2. **提出一种领域知识辅助约束的机器学习方法。**由于一阶逻辑表达的领域知识较为复杂，常规的机器学习方法难以使用它们来对机器学习的优化过程进行约束。本文提出的 LASIN 方法可以将领域知识转化为机器学习中关于假设模型搜索空间的约束条件，以要求从数据中学得的模型与领域知识保持一致。实验表明，与未引入领域知识的机器学习方法相比，LASIN 方法学得的模型拥有更好的泛化性能。
3. **提出一种机器学习驱动的领域知识精化方法。**知识精化的目标是从数据中学习具有良好可理解性的一阶逻辑规则，以往的知识精化方法难以从不含逻辑符号的数据中进行学习。本文提出的 KRL 方法将知识精化问题转化为机器学习任务，它首先使用机器学习模型将样本转化为逻辑符号，并使用这些逻辑符号增广领域知识，最终对增广后的领域知识进行精化。在计算机视觉任务中的实验结果表明，KRL 能够学习出目标概念的形式化定义。

4. **提出一种领域知识与机器学习互促结合框架。**将数据驱动的机器学习与知识驱动的逻辑推理相结合一直是人工智能中的核心挑战。本文提出的反绎学习是一种将此二者进行互促结合的框架，它既能将领域知识转化为训练机器学习模型所需的样本标记，又能利用机器学习将数据转化为增广和精化领域知识所需的逻辑符号。基于该框架，本文提出一种结合知识精化与深度学习的算法 NLM，并在一类复杂学习任务中验证了它的有效性。

关键词：人工智能；机器学习；知识处理；一阶逻辑；知识精化；统计学习；深度学习；反绎学习

南京大学研究生毕业论文英文摘要首页用纸

THESIS: Research on Integrating First-Order Logical Domain Knowledge with Machine Learning

SPECIALIZATION: Computer Science and Technology

POSTGRADUATE: Wang-Zhou Dai

MENTOR: Professor Zhi-Hua Zhou

Abstract

There usually exists large amount of domain knowledge in practical machine learning tasks, and the knowledge can be easily expressed by first-order logic language. This dissertation tries to investigate: 1) how to exploit such kind of domain knowledge to improve machine learning; 2) how to use machine learning to help with knowledge refinement; 3) how to integrate knowledge-driven reasoning and data-driven machine learning in a mutually beneficial way. Particularly, four machine learning approaches and frameworks are proposed to answer the above three questions:

1. **A machine learning approach that augments training examples with domain knowledge.** It is difficult for traditional machine learning methods to utilize information expressing relations between objects. This kind of information can be conveniently formalized by first-order logical domain knowledge. In order to exploit this kind of domain knowledge for improving the performance of machine learning, we propose the SUL (Statistical Unfolded Logical) learning approach, which transforms the domain knowledge into augmented features that could be directly exploited by machine learning algorithms. Experimental result shows the effectiveness of SUL learning against those learning approaches that do not consider domain knowledge.
2. **A machine learning approach that exploits domain knowledge as constraints.** Because of the highly complexity of first-order logical domain knowledge, it is hard for ordinary machine learning techniques to use them to constrain the learning process. We propose the LASIN (Logical Abduction and Statistical INduction) method to transform the domain knowledge into constraints on hypothesis space of machine

learning tasks. Therefore, LASIN can force the models learned from data to be consistent with the domain knowledge. Experimental results show that, comparing to the approaches that do not introducing first-order logical domain knowledge, the models learned by LASIN can generalize better in the testing data.

3. **A knowledge refining approach that is driven by machine learning techniques.** The target of knowledge refinement is to learn comprehensible logical rules from data. Traditional knowledge refining approaches can not be applied on the data that is not formulated by logical symbols. We propose the KRL (Knowledge Refinement by Learning) method to transform knowledge refinement problems into machine learning tasks. KRL first augments the domain knowledge by using machine learning models to extract primitive logical facts, then it can apply machine learning techniques for learning first-order logical rules from the augmented domain knowledge. Results on the experiments of computer vision tasks show that KRL is able to learn formal definition of the target concepts from images.
4. **A framework that integrates domain knowledge and machine learning in a mutually beneficial way.** Integrating knowledge-driven logical reasoning and data-driven machine learning has been a key challenge in Artificial Intelligence area for a long time. In this dissertation, we propose the Abductive Learning framework to combine them in a mutually beneficial way. On the one hand, Abductive Learning transforms the domain knowledge into supervision information that can be used for training the machine learning model; on the other hand, it uses machine learning to extract primitive logical facts that augment domain knowledge for logical reasoning and knowledge refinement. Based on this framework, we further proposed the NLM (Neural Logical Machine) algorithm that combines knowledge refinement and deep learning, the experimental results on complex learning tasks shows its effectiveness.

keywords: Artificial Intelligence, Machine Learning, Knowledge Processing, First-Order Logic, Knowledge Refinement, Statistical Learning, Deep Learning, Abductive Learning

目　　次

目　　次 …… v

1　绪论 …… 1

1.1　引言 …… 1

1.2　研究背景 …… 3

1.3　有待研究的问题 …… 14

1.4　本文工作 …… 15

2　一种领域知识增广样本的机器学习方法 …… 17

2.1　引言 …… 17

2.2　SUL 学习 …… 18

2.3　实验验证 …… 22

2.4　小结 …… 31

3　一种领域知识辅助约束的机器学习方法 …… 33

3.1　引言 …… 33

3.2　LASIN 方法 …… 34

3.3　实验验证 …… 38

3.4　小结 …… 48

4　一种机器学习驱动的领域知识精化方法 …… 51

4.1　引言 …… 51

4.2　KRL 方法 …… 52

4.3　实验验证 …… 62

4.4　小结 …… 72

5　一种领域知识与机器学习互促结合框架 …… 73

5.1　引言 …… 73

5.2　反绎学习 …… 74

5.3 实验验证 ····· 85
5.4 小结 ····· 93

6 结束语 ····· 95

参考文献 ····· 97

致 谢 ····· 115

A 攻读博士学位期间的学术成果和获奖情况 ····· 117

第一章　绪论

1.1　引言

人工智能（Artificial Intelligence，AI）是研究如何使机器像人一样思考、行动并基于已知条件采取合理行为的科学[146]，它起源于 1956 年在达特茅斯学院召开的夏季研讨会。如今，人工智能已经发展为一个以计算机科学为基础，兼及数学、哲学、控制论、神经科学、认知心理学、语言学和伦理学等广泛学科的前沿领域。经过 60 多年的演进，它已逐渐成为新一轮科技革命和产业变革的核心驱动力，我国国务院也于 2017 年印发的《新一代人工智能发展规划》明确指出人工智能将成为国际竞争的新焦点、经济发展的新引擎和社会建设的新机遇，将人工智能发展作为重要国家战略。

早期人工智能主要研究基于自动逻辑推理的问题求解（problem-solving）[119]，对数学定理进行自动证明是当时的热点之一[52,141]，我国著名数学家吴文俊院士和张景中院士曾在该领域中做出突出贡献[18,179]。此类研究的目标是设计一种通用的推理机制，以求解任意表示为一阶逻辑形式的问题[118]。但是，尽管它们通用，却难以被应用在大规模的实际问题中[146]。上世纪 70 年代兴起的专家系统（Expert Systems）[71] 将人工智能带入了“知识期”，它们通过预设更强且更加领域相关的知识来实现大规模的推理。当逻辑程序设计（Logic Programming）[4,86] 和 Prolog 语言[14] 出现后，专家系统拥有了更强的表达能力和推理能力，并在当时成为人工智能的代名词，逐渐形成一个规模庞大的产业[146]。

从上世纪 80 年代中期开始，人工智能研究的重心逐渐转向机器学习[13,180]领域。这是因为专家系统所依赖的领域知识不仅成本较高，许多任务还由于专家知识难以总结、或专家不愿分享知识而难以完成，形成了所谓的“知识工程瓶颈”[180]。因此，机器学习致力于研究如何通过计算的手段，利用经验来改善系统自身的性能。具体来说，它通过学习算法（learning algorithm）从数据中产生模型（model），以对未见过的数据进行相应判断。如今，机器学习已经成为人工智能的核心研究领域。

随着机器学习在实际问题中的应用愈来愈广泛，人们对其可理解性和可重用性的要求愈来愈高[177]。许多学者认为，解决这两个问题的一种可行策略便是将机器学习与一阶逻辑表示的知识相结合，使机器学习能够引入人类所拥有的领域知识，并将学得的模型转化为人类可理解的规则[103,145,146,180]。然而，目前处于主流的绝大多数统计机器学习[62]与深度学习[55]方法却很难对此类知识加以利用。这在一定程度上阻碍了人工智能领域的发展。

利用一阶逻辑领域知识辅助机器学习的方法主要包括统计关系学习（Statistical Relational Learning，SRL）[54]与概率归纳逻辑程序（Probabilistic Inductive Logic Programming，PILP）[32,34]。SRL 能够为概率图模型（Probabilistic Graphical Model，PGM）[83]赋予一阶逻辑语义，PILP 则选择在逻辑程序[4,86]的基础上引入概率分布[120,147,152]再进行学习。这两种方法均能够将一阶逻辑表示的领域知识作为概率模型的结构，再通过机器学习从数据中估计概率分布的参数。然而，这两类学习方法只能被应用在基于逻辑表示的问题中，即输入数据为具体逻辑事实（ground facts）的关系型学习（relational learning）[40]任务中。近年来，还出现了一些将一阶逻辑知识与深度神经网络等其他机器学习算法相结合的方法[2,37,102,143,155]，但它们大多只能引入一些形式简单、表达能力有限的一阶逻辑知识。

另一方面，人工智能领域中也存在使用机器学习进行领域知识精化的研究。知识精化的目标是使用训练数据将大量的领域知识归纳为少数几条人类可理解的一阶逻辑规则，以对目标概念进行定义。此类方法大多是为了解决“知识工程瓶颈”而提出，其主要代表为决策树（Decision Trees）[15,137]、命题规则学习（Propositional Rule Learning）[50]和归纳逻辑程序设计（Inductive Logic Programming, ILP）[93,107]等。其中决策树和命题规则学习只能学习表达能力十分有限的命题逻辑规则，但它们的技术简单实用，其中决策树到今天仍是最常用的机器学习技术之一。ILP 是基于一阶逻辑表示的学习方法，它拥有更强的表达能力，能够从数据中学习出形式更加复杂的知识，因此学习起来更加困难。作为 ILP 的概率扩展，PILP 同样可以使用机器学习方法实现知识精化，但由于它需要同时学习逻辑规则和概率分布参数，复杂度比 ILP 更高。此外，这些方法只能从逻辑符号表示的数据中进行知识精化，应用范围有限[145]。

将数据驱动的机器学习与知识驱动的逻辑推理相结合，是人工智能领域中的核心挑战之一[103,145,178,180]。长期以来，关于二者的研究绝大多数是分离的，上文所提到的方法也只能被应用在有限的场景下。而如何将此二者纳入同一框

架，实现学习过程中的互相促进、共同提升，则更少有研究。本文对将一阶逻辑领域知识与机器学习相结合问题分为三种情况：1) 利用领域知识辅助机器学习；2) 利用机器学习促进知识精化；3) 将领域知识与机器学习互促结合。为了对这些问题进行探索，本文提出了四种机器学习方法和框架作为创新成果。

本章后续内容中将首先简要介绍研究背景，然后提出一阶逻辑领域知识与机器学习相结合时有待研究的问题，最后介绍论文的组织。

1.2 研究背景

如何将领域知识与机器学习相结合一直是人工智能领域的核心问题之一。机器学习中使用的领域知识有多种表示形式，它可以是优化问题中对目标函数和假设空间的约束、通过函数表达的正则化项、概率图模型的结构、甚至学习算法的选取等等。本文主要关注的是由一阶逻辑语言表示的一般领域知识，它比以上形式的领域知识拥有更丰富的表达能力，且对于人类用户而言具有更佳的可理解性。本小节将先简单介绍一阶逻辑语言，然后对使用领域知识辅助机器学习和使用机器学习进行领域知识精化的研究现状作简单综述。

1.2.1 一阶逻辑语言

一阶逻辑（First-Order Logic）是一种形式语言（formal language）[43]。它的表达能力非常强，因此能够被用于描述形式较为复杂的一般领域知识。

一阶逻辑语言的构成元素有*连词*（“$\wedge$”、“$\vee$”、“$\neg$”和“$\leftarrow$”）、*常量*、*变量*、*量词*（“$\forall$”和“$\exists$”）、*谓词*以及*函数*。在不做特别说明的情况下，本文中将省略所有*一阶逻辑公式*（formulas）中的全称量词“$\forall$”，并以小写英文字母开头的词代表谓词、函数和常量，以大写英文字母开头的词代表变量或任意一阶逻辑公式。

机器学习中的领域知识通常由*一阶逻辑规则*（first-order logical rules）构成的集合表示。这些规则是一类特殊的一阶逻辑公式，它们的形式为：

$$A \leftarrow B_1 \wedge B_2 \wedge \ldots \wedge B_m. \tag{1.1}$$

其中蕴涵符“$\leftarrow$”左边的 A 被称为*规则头*（head），表示规则的结果。它是一个*原子公式*（atom），即形如 $t(X_1, X_2, \ldots, X_n)$ 的*项*（term）。其中 t/n 为该

原子公式的谓词，n 被称为*元数*（arity），它代表谓词 t 中参数的个数。式1.1“$\leftarrow$”左边的部分被称为*规则体*（body），表示此规则的前提。它是有有限个*逻辑文字*（literals）的合取，其中每个逻辑文字 B_i 是原子公式或它的否定形式。规则体中文字的个数 $m \geq 0$ 被称为规则的长度。当 $m = 0$ 时规则退化为一个*事实*（fact）“$A \leftarrow$”，它表示在领域知识中 A 的赋值恒为真，通常情况下可省略其右边的蕴涵符并将其表示为“A”。当一阶逻辑规则中不存在变量时，它被称为*具体的*（grounded），具体的原子公式被称为*具体事实*（ground facts 或 groundings）。

一阶逻辑系统通过*赋值*（assignment）来计算每个公式的真假。最基本地，它可对每个具体事实进行赋值，令其要么为 *true*，要么为 *false*。在人工智能领域所使用的一阶逻辑系统通常采用*闭世界假设*（closed world assumption，CWA），即未出现在领域知识中的具体事实赋值均为 *false*，一般通过 NAF（negation as failure）[20] 实现。当所有具体事实的赋值确定后，一阶逻辑会按照一定规则对函数与谓词进行演算，以得到其它公式的真值。如果存在一种赋值方使得公式的赋值为 *true*，则称此公式为*可满足的*（satisfiable）。公式（集）之间的可满足性被叫做*语义蕴涵*（entailment），记为“$\Gamma \models T$”，它表示一切令 Γ 为真的赋值也使得 T 为真；公式（集）之间的*证明关系*被称为*形式蕴涵*（syntactic consequence）关系，记为“$\Gamma \vdash T$”，一般读作“Γ *可推出*（yields）T”，它表示 Γ 可以凭借一系列公理证明出 T。对于一阶逻辑系统来说，可满足与可证关系是等价的[43]。

在人工智能领域中应用最常见的一阶逻辑系统是由 Prolog 语言[14] 实现的逻辑程序（Logic Programs）[4,86]。它采用基于*归结原理*（resolution principle）[141] 的 SLD 归结（Selective Linear Definite clause resolution）[87] 方法对形如1.1的规则进行推理和演算。与采用 if-then-else 规则进行知识表示方式相比，基于一阶逻辑语言的逻辑程序能够表示更复杂的递归规则。长期以来，它被大量应用在专家系统和机器学习等领域中[21,32,34,97,107,114,156]。

1.2.2 引入一阶逻辑领域知识的机器学习方法

人工智能领域中已存在一些引入领域知识的学习方法，它们主要包括统计关系学习（SRL）[54] 和概率归纳逻辑程序设计（PILP）[32,34]。近年来，研究者们还提出了一些在深度神经网络等机器学习模型中引入一阶逻辑领域知识的方法[2,37,102,143,155]。

由于 PILP 是归纳逻辑程序设计（ILP）基于概率的扩展，它更重要的功能是使用机器学习方法对领域知识进行精化，因此我们将在第1.2.3小节中对其进行介绍。下面将仅介绍其余两类引入一阶逻辑知识的机器学习方法。

统计关系学习

SRL[54] 试图为概率模型[83] 赋予一阶逻辑语义，这种引入语义的方法一般被称为“知识库模型构建”（Knowledge Base Model Construction，KBMC）[3,169]。KBMC 将一阶逻辑中的具体事实当作布尔随机变量，并将概率分布定义在领域中全部可能的具体事实上。对全体逻辑事实的一种赋值一般被称为一个可能世界（possible world）[122]。也就是说，KBMC 将领域中全部随机变量的联合概率分布定义在所有的可能世界中。此外，KBMC 把一阶逻辑逻辑公式中的关系解释为随机变量间的依赖性，将逻辑推理直接更改为概率推断。因此，SRL 很难利用一阶逻辑演算的特性进行推理或学习，领域知识中的一阶逻辑规则必须彻底转化为概率模型才能使用。

最简单的一种 KBMC 方式是直接将逻辑规则中的蕴涵符号解释为条件概率，此类 SRL 模型一般将不含递归规则的逻辑程序解释为一个贝叶斯网（Bayesian Network）[83]。例如 Horsch 和 Poole[65] 提出的一种一阶贝叶斯网、Friedman 等人[49] 提出的 PRM（Probabilistic Relational Model）、Costa 等人[25] 提出的 CLP(BN) 和 Keristing 等人[74] 提出的 BLP（Bayesian Logic Programs）等 SRL 模型。

为了令 SRL 能够表达递归规则或循环规则，研究者们将无向图和有向有环图引入了 SRL 模型。Taskar 等人[157] 提出的关系型马尔可夫网（Relational Markov Networks，RMN）与 Richards 和 Domingos[140] 提出的马尔可夫逻辑网（Markov Logic Network，MLN）均试图使用马尔可夫网（Markov Network）[83] 表示一阶逻辑规则；Neville 和 Jensen[123] 提出的 RDN（Relational Dependency Network）则将一阶逻辑规则表示为依赖网（Dependency Networks，DN）。此类 SRL 模型均无法进行精确概率推断（exact inference），只能通过吉布斯采样（Gibbs sampling）或变分法（variational methods）做近似推断[83]。

SRL 使用的模型和引入的领域知识均为带权一阶逻辑公式，它的学习任务可分为参数学习和结构学习。与传统的概率图模型类似，SRL 的参数学习大多可通过（Maximum Likelihood Estimation，MLE）和最大化期望（Expectation－Maximization）算法解决[49,74,84]。但对基于无向图模型的 SRL 模型来说，我

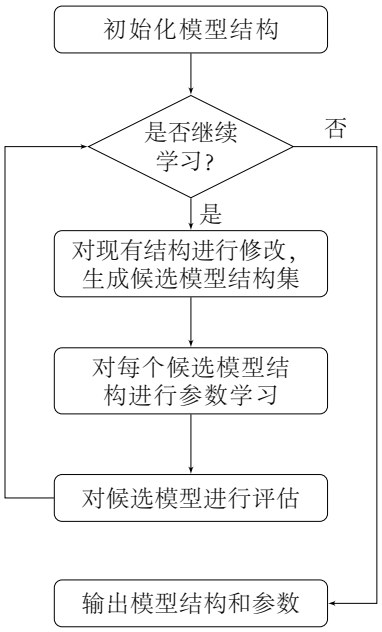


图 1.1 SRL 与 PILP 模型的结构学习流程

们无法写出其概率似然的闭式形式，因此一般先使用伪似然函数（pseudo-likelihood）[83] 以对其概率似然进行近似，然后通过梯度下降进行优化。例如，Richardson 和 Domingos[140] 提出采用 L-BFGS 对 MLN 的参数进行优化，Lowd 和 Domingos[99] 对该方法进行了改进。Hyuhn 等人[67,68] 则提出使用 DPM（Discriminative Probabilistic Model）[157] 中的最大间隔（max-margin）方法学习 MLN 的参数。此外，对于某些结构形式非常简单的模型，SRL 还可以利用提升推断（lifted inference）[133] 来提高参数学习的效率[77,162]。

大部分 SRL 结构学习算法均遵循图1.1所展示的流程。其中基于有向图的 SRL 结构学习算法与贝叶斯网的结构学习方法基本相同[49,53]。而基于无向图的 SRL 模型结构学习的研究一直是 SRL 领域的热点。Kok 和 Domingos[80] 提出通过集束搜索（beam search）生成候选模型结构，并使用带权伪似然（weighted pseudo-likelihood，WPLL）对候选模型进行评估。Biba 等人[12] 则使用判别式方法学习 MLN 的参数，并在生成候选结构时加入随机扰动以跳出局部最优。Mihalkova 和 Mooney[104] 试图通过自底向上的搜索方法限制 MLN 结构的搜索空间，Kok 和 Domingos 提出的 LHL（Learning via Hypergraph Lifting）[81] 和 LSM（Learning using Structural Motif）[82] 则通过预先对训练样例聚类进一步缩小搜索空间。

为了解决结构-参数交替优化效率低下的问题，Huynh 和 Mooney[66] 提出的

判别式 MLN 结构学习方法先使用 ILP 生成大量一阶逻辑规则结构，然后在参数学习的优化过程中使用 L_1 范数对非 0 权重进行惩罚，以尽可能多地删除规则。虽然它可以同时学习参数并删除规则，但他无法对规则进行增加或修改，因此许多情况下效果不如基于交替优化的方法。Natarajan 等人提出的 RDN-Boost[115] 则试图使用函数梯度下降（functional gradient descent）同时 RDN 的结构和参数，并在实验中取得了很好的效果。Khot 等人提出的 MLN-Boost[75] 将该方法推广至 MLN 的结构学习，不过由于 MLN 的联合概率分布比 RDN 更复杂，因此它依然需要交替地学习结构与参数。

SRL 能够引入任意形式的一阶逻辑规则，其中 MLN[140] 所使用的一阶逻辑规则形式甚至比逻辑程序[4,86] 更加丰富。但是，此类模型的复杂度较高，学习难度较大。一方面，概率图模型中的概率推断和结构学习问题均至少是 NP-难的[83]，因此以它为基础的 SRL 亦拥有同样的复杂度。另一方面，它们所构成的概率图模型中每个节点均为具体逻辑事实，因此由一阶逻辑领域知识转化而得的 SRL 模型规模往往十分庞大，这进一步增加了 SRL 的难度。此外，一旦领域知识无法精确地描述领域中隐含的依赖关系结构，通过它转化而得的概率图模型结构同样会不够准确，这将影响参数学习的性能[83]。

SRL 一般只能被应用在关系型学习[40] 问题中，此类学习问题的输入数据均为一阶逻辑公式或逻辑事实。然而，大部分机器学习问题中所使用的往往是像素表示的图像、视频和波形表示的声音等不存在逻辑符号的原始数据。Russell 曾指出该方面研究的空缺[145]，他评论道：“真实世界中的物体不会像戏剧角色一样（用符号）给自己标注一个独特的身份再出场”，这条断言也在 SRL 领域中引起了热烈讨论[16]。简而言之，SRL 方法难以被应用于一般的机器学习问题。

一阶逻辑领域知识在其他机器学习方法中的应用

Towell 和 Shavlik[160] 提出的 KBANN（Knowledge-Based Artificial Neural Networks）以及 Hölldobler 和 Kalinke[64] 提出的 PCMLP（Parallel Computational Model for Logic Programming）试图用命题逻辑规则初始化神经网络结构，然后再使用该网络进行机器学习。França 等人[48] 提出的 CILP++ 则对这种学习模式进行扩展，它将一阶逻辑领域知识转化为命题逻辑规则再用类似方式进行学习。不过这些方法只能构建单隐层神经网络，无法引入更复杂的知识。

近年来，一些研究者们也试图利用一阶逻辑规则来对深度神经网络结

构进行初始化[37,51,143,155]。此类方法均试图将逻辑符号表示为嵌入式形式（embedding）并使用向量乘法计算逻辑事实的赋值，以达到将逻辑推理转化为深度神经网络计算的目的。除了直接使用规则本身初始化网络结构，Rocktäschel 与 Riedel[142] 还提出用逻辑程序在后向链（backward chaining）推理[14] 时形成的搜索分支结构以对网络进行初始化。以上基于嵌入式表示的方法泛化能力有限，当测试数据中出现训练数据中不存在的逻辑符号时便无法使用。Cohen 等人[23,173] 提出的 TensorLog 不再使用嵌入式表达，但它只能处理一类结构非常特殊的规则。Evans 和 Grefenstette[44] 提出的 ∂ILP 则将保留了基于二阶反绎推理的一阶逻辑归纳的结构，将每个一阶逻辑公式当做连续函数进行学习。∂ILP 在一些简单 ILP 任务中甚至可以取得与 Metagol[114] 相近的效果，但它只能学习由两条规则构成假设模型，因此表达能力非常有限。

还有一些研究者们尝试通过端到端（end-to-end）学习，将机器学习问题中的逻辑推理完全替换为神经网络中的可微分函数运算[57,58,170]。这一类方法试图将对领域知识利用和机器学习过程全部融入深度神经网络，然后学习一个从样本空间直接到最终目标概念的映射。它们在一些较为简单的特殊任务中取得了一定效果，且甚至能够引入自然语言表达的领域知识[57,170]。

事实上，以上这些基于深度神经网络的方法并未完全实现一阶逻辑知识与机器学习的结合，因为这些模型中均无法实现真正的一阶逻辑推理，并且它们输出的模型仍然是不具有可理解性的黑箱模型。此外，受限于神经网络模型的表达能力，大部分模型只能够引入形式十分简单的领域知识，例如大小有限且长度有限的规则集[44]、由特殊形式谓词和特殊结构规则构成的领域知识[23,142] 等等。而端到端学习则难以在训练样本有限的情况下实现复杂问题中的逻辑推理，例如对长文本进行阅理解等等[72]。

除了神经网络模型，还有一些研究尝试在统计学习模型中引入一阶逻辑语言表达的领域知识。例如，Andrzejewski 等人[2] 和 Mei 等人[102] 分别提出在 LDA（Latent Dirichlet Allocation）模型的先验分布和后验分布中引入逻辑规则作为约束。它们均采用类似 SRL 的方式，将一阶逻辑规则转化为概率模型以分别对样本和标记分布进行约束。但是与上面基于神经网络的方法一样，它们只能对一些形式较为简单的领域知识加以利用，并且仅能应用在已被表示为逻辑符号的自然语言文本数据中。

1.2.3 基于机器学习的一阶逻辑知识精化方法

为了通过机器学习方法实现知识精化——即从领域知识和训练数据中学习一阶逻辑规则，研究者们分别提出了归纳逻辑程序设计（ILP）[93,107]和概率归纳逻辑程序设计（PILP）[32,34]，本小节将分别对它们进行介绍。

归纳逻辑程序设计

归纳逻辑程序设计（ILP）是机器学习与逻辑程序设计的交叉领域，它特指以一阶逻辑归纳理论（First-Order Logical Induction Theory）[121]为基础，并以一阶逻辑规则为模型的机器学习方法，它的任务是对一阶逻辑表示的知识进行精化。具体来说，ILP 的目标是从逻辑程序表示的背景知识（background knowledge）中自动归纳出一个由一阶逻辑规则构成的假设模型（hypothesis），以使得该假设模型与背景知识一起能够满足关于所有目标概念（target concept）的样例（examples）。

ILP 命名中的“归纳”（induction）是逻辑中“演绎”（deduction）推理的逆过程。因此 ILP 中的大部分方法均试图将一阶逻辑中的演绎理论进行逆转以进行归纳学习。上世纪 70 年代已出现一些关于一阶逻辑归纳的先驱性工作[128–130,135,167]，它们对一阶逻辑归纳理论的探索奠定了 ILP 的基础。其中 Plotkin[128]提出的最小一般泛化（least general generalization，LGG）是最早的一阶逻辑归纳运算，它与自动定理证明中常用的归结原理[141]的最一般合一（most general unification）互为逆运算，并成为日后 ILP 中最基础的泛化运算之一。

基于 LGG 运算最具代表性的 ILP 方法是 Muggleton 和 Feng[113]提出的 Golem。不过，LGG 只能对已有的逻辑规则进行泛化，而不能从仅拥有逻辑事实的背景知识中学习新规则。在这种情况下，Golem 采用了 Plotkin[129]提出的相对最小一般泛化（relative least general generalization，RLGG）方法进行学习。它首先从背景知识中生成一个由具体逻辑规则构成的饱和集（saturation）[144]，然后对它们进行最小一般泛化以学习一阶逻辑规则。对于简单的知识精化问题，基于 LGG 和 RLGG 的 Golem 有着较好的理论性质。当背景知识和样例均由具体事实构成时，RLGG 可以得到所有符合 ILP 目标的一阶逻辑规则[130]。不过，当背景知识中存在形式较复杂的一阶逻辑公式，或目标概念必须通过多条规则才能表达时，仅仅基于 RLGG 进行归纳将无法学得正确的模型[130]。

对于更复杂的知识精化问题，则可以通过逆归结（inverse resolution）[106]方法进行学习。基于该方法的代表性 ILP 系统有 Duce[105] 和 Cigol[112]，它们分别被用于命题规则学习和 ILP 任务中，其中后者第一次实现了一阶逻辑归纳中的谓词发明（predicate invention）[88]。逆归结方法主要包括吸收、辨识、内构和互构四种运算，它们分别属于归结原理[141] 四种特殊情况下的逆运算。其中内构和互构可以自动构造出背景知识里未事先定义的概念。谓词发明可以对背景知识中定义的领域结构进行补充，并帮助用户对领域建立新的认识。基于逆归结的 ILP 方法能够学习更加复杂的概念，不过它的复杂度极高，在实际任务中很难应用[113]。

基于 LGG、RLGG 和逆归结的一阶逻辑归纳不满足完备性，即对于形式不受限制的目标概念，它们无法找到所有满足此目标概念的假设模型。换而言之，真实模型（ground-truth），或与真实模型等价的最优假设模型（optimal hypotheses），很可能不存在于以上 ILP 方法的假设空间中。具体来说，当目标概念中存在递归规则时，这些方法均无法学得正确的模型。这是因为它们所采用的基础归纳运算 LGG 并不是演绎推理的完备逆运算[56]。为了解决该问题，Muggleton[108] 提出了逆语构蕴涵（inverse implication）方法。这是一种完备的一阶逻辑归纳运算，理论上它可以解决 ILP 对递归规则的学习问题。但可惜的是，逆语构蕴涵中有一个子问题是不可判定（undecidable）[43] 的，因而它并未形成有效的算法。目前为止，基于逆语构蕴涵思想的 ILP 方法仅有 Idestam-Almquist[69] 在对其进行放松后提出的逆 T-语构蕴涵。

以上结果表明对演绎推理直接进行逆转会带来许多麻烦，因此，Muggleton[109] 提出应当从模型可满足性的角度来研究一阶逻辑归纳理论。他利用“逆否命题与原命题在语义上等价”这一性质，巧妙地将一阶逻辑归纳问题转换为一阶逻辑演绎问题，并在该理论的基础上提出了 Progol[109] 系统。该方法首先将所有训练样例取反以得到它们的否定形式，再从背景知识和否定形式的训练样例出发进行演绎推理，最后只需将推导出的结果取反，即可得到原归纳问题中所求解的假设模型。理论上，Progol 的推理能够保证归纳的完备性，不过它要求演绎推理所得的具体事实形式足够丰富。因此在算法实现中，Progol 会借助先验知识对演绎推理过程进行约束。此外，它还引入了精化算子（refinement operator）[121] 对归纳过程进行剪枝，以提高知识精化的效率。

基于命题化（propositionalization）[47,89,90,94] 的 ILP 方法是一种较为特殊的一阶逻辑归纳方法。它们试图将逻辑程序表示的领域知识和训练样例转化为一

个属性-值（attribute-value）数据集，然后通过命题规则学习[50]方法从中学得一阶逻辑表示的假设模型。受表达能力的限制，基于命题化的 ILP 方法一般被应用于以个体为中心的领域[90]，即目标概念只描述个体属性而不涉及个体间关系的领域。

除了监督学习，ILP 也可以被用在其他类型的机器学习问题中。De Raedt 与 Bruynooghe[31] 提出的 CLINT 是一个基于主动学习的 ILP 系统。它能够向用户询问两种问题：一种会询问关于目标概念的具体事实是否为真；另一种会自动则给出一个关于目标概念的原子公式，并请求用户将其中的未知变替换为符合目标概念的常量。在询问次数充足时 CLINT 对于一阶逻辑归纳完备。

以上 ILP 方法均试图在一阶逻辑系统中实现领域知识精化，但它们均未完美地解决递归规则学习和谓词发明问题。Lloyd[98] 指出许多一阶逻辑归纳中的问题可以在高阶逻辑（Higher-Order Logic）系统[43] 中解决。近年来，Inoue 等人[70] 首次提出在二阶逻辑（Second-Order Logic）系统中使用反绎推理（abduction）进行一阶学习。该方法后来被 Muggleton 等人扩展为更一般的元解释学习（Meta-Interpretive Learning，MIL）[114] 并实现为 Metagol 算法，它更好地解决了一阶逻辑归纳中的递归规则学习与谓词发明。Metagol 实现了一个二阶元解释器（meta-interpreter）以尝试对符合目标概念的训练样例进行证明。在此过程中，它会自动调用事先定义的二阶规则模板生成一阶逻辑规则作为假设模型，以完成对样例的证明。实验表明，Metagol 比以往的 ILP 方法更适合于对包含递归关系的领域知识进行精化[114]。

作为一种机器学习方法，ILP 的 PAC 可学习性[161] 亦是研究者们普遍关注的问题之一。在一部分形式受限制的假设空间中，ILP 是 PAC 可学的。Page 和 Frisch[126] 证明了单条不含函数符且非递归的一类特殊一阶逻辑公式的 PAC 可学习性，Džeroski 等人[41,42] 和 Cohen[22] 则证明了一些结构较简单的一阶逻辑规则集 PAC 可学。不过，对于规则形式不受限制的一般一阶逻辑规则集，ILP 不是 PAC 可学习的[76]。这是因为 ILP 在学习过程中必须不断对学得的一阶逻辑规则进行冲突检测，而这是个 NP-难问题。以上结果均表明，由于假设空间过于复杂，一阶逻辑下的知识精化是一类非常复杂的学习问题[24]。

由于一阶逻辑规则的结构是离散的，大部分 ILP 方法均采用贪心或启发式搜索进行一阶逻辑规则学习。因此 ILP 的效率往往非常低，并且很难考虑规模较大的领域知识和训练样例。此外，ILP 方法与 SRL 类似，只能对已经表示为逻辑符号的领域知识进行精化，无法应用在更一般任务中。

概率归纳逻辑程序设计

基于一阶逻辑表达的 ILP 一般只能在不含噪声的领域中进行学习，然而大部分实际问题中收集的数据甚至领域知识均可能含有噪声。为了在此类领域中进行知识推理与精化，研究者们提出了概率逻辑程序（Probabilistic Logic Programs，PLP）[29,120] 以及基于它的机器学习方法——概率归纳逻辑程序设计（PILP）[34]。

为了在逻辑系统中引入不确定性，Nilsson[122] 提出的概率逻辑（probabilistic logic）首先将命题逻辑系统中的概率分布定义在可能世界中。随后，Dastin[29] 将 Nilsson 的概率逻辑扩展至一阶逻辑程序并首次提出了 PLP 的概念和推理方式。经过多年来大量研究者们的不懈努力[131,132,165,166,168]，PLP 的语法不断演进，逐渐成为了一类成熟的形式系统。

与 SRL 类似，PLP 将概率分布定义在领域内所有具体事实上，它将每个具体事实作为一个布尔随机变量，并称为*概率事实*（probabilistic facts），一般表示为如下形式：

$$w :: A. \tag{1.2}$$

它表示逻辑事实 A 赋值为 *true* 的概率为 w，即 A 服从参数为 w 的伯努利分布。PLP 中的逻辑公式以通过类似的符号赋权：

$$w :: A \leftarrow B_1 \wedge B_2 \wedge \ldots \wedge B_m. \tag{1.3}$$

这种规则被称为*赋权逻辑规则*或*概率逻辑规则*，它表示当 $\{B_1, \ldots, B_m\}$ 赋值均为 *true* 时，A 服从参数为 w 的伯努利分布。

PLP 中的概率分布等价于所有概率事实的联合分布，即可能世界的分布，这种定义方式被称为分布语义（distribution semantics）[147]。一旦可能世界确定，PLP 便退化为一个普通的逻辑程序，它可以进行正常的一阶逻辑推理。

现在应用最广泛的 PLP 系统是 De Raedt 等人提出的 ProbLog[33]，它集成了许多成熟的 PLP 语法，实现了高效的概率逻辑推理。推理开始前，ProbLog 会将 PLP 编译为二元决策图（Binary Decision Diagrams，BDD）[1,38]，将 PLP 的概率推断转化为一类可满足性（SAT）问题。因此，ProbLog 可使用带权模型计数（weighted model counting, WMC）实现高效的概率逻辑推理。它的最新版本 ProbLog2[39] 将基于 BDD 的 WMC 改进为基于 d-DNNF（deterministic Decomposable Negation Normal Form）[30] 的 WMC，进一步提高了推理效率。

概率归纳逻辑程序设计（PILP）[34] 是以 PLP 为模型的一类机器学习方法。因为 PLP 允许逻辑事实依概率为真，因此 PILP 不再要求学得与背景知识和训练样例完全一致的假设模型，仅要求假设模型尽可能在对训练样例赋值的判断上少犯错误。换而言之，PILP 的目标是学习一个假设模型，以最大化训练样例在该模型和背景知识中的后验概率。其中，待学习的假设模型是由概率逻辑公式（probabilistic logic rules）组成的 PLP，它既包含了一阶逻辑规则的结构，也包含了概率分布的参数。因此，PILP 的任务与 SRL 类似，可分为参数学习和结构学习[34]。由于引入了概率分布，所以 PILP 使用的训练样例与 ILP 中的稍有区别，它们的本质是关于逻辑事实在可能世界中赋值的采样。为了能够用它们对模型进行估计，一般需要假设这些采样服从独立同分布。

PILP 的参数学习是在领域知识中逻辑规则结构确定时下对概率分布的参数进行估计。这些参数一般是逻辑事实或一阶逻辑公式为真的概率值。若训练数据中的每次采样均包括了领域中的全部具体事实，参数学习可以通过 MLE 求解。例如 Cussens[27] 提出的 FAM 方法和 ProbLog 中使用的 LFI-ProbLog[60] 方法等等。此外，在一类特殊的数据集中，也可以采用梯度下降[59] 方法以进行 PILP 的参数学习。

在许多任务中，训练样例所包含的逻辑事实可能不完全。这种情况下的参数学习相当于在有隐变量的情况下估计概率分布的参数。此类问题一般可用 EM 算法进行求解。Sato[147–149] 提出的 PRISM 是最早提出使用 EM 进行参数估计的 PILP 方法，前文介绍的 LFI-ProbLog[60] 同样可以通过 EM 对 ProbLog 进行参数估计。早期的 PILP 方法均无法对包含递归规则的 PLP 进行参数学习，不过在最新的 ProbLog2[39] 中，LFI-ProbLog 在此类 PLP 上实现了近似的参数估计[45]。为了对 PILP 的 EM 进行加速，Bellodi 和 Riguzzi[7] 提出的 EMBLEM（EM over BDDs for probabilistic Logic programs Efficient Mining）可提前将训练样例编译为 BDD 以提升 EM 中概率推断的效率。

PILP 的结构学习则能够进行知识精化，它既需要学习一阶逻辑规则结构，又需要对学得的 PLP 进行参数估计，这是一个 NP-完全问题[34]。大部分 PILP 结构学习算法也遵循图1.1中所示的流程，即交替地进行一阶逻辑规则结构搜索和参数学习。其中修改模型结构的方式可以是对一阶逻辑规则进行增删或改变某些规则中的逻辑文字。

Muggleton[110,111] 提出的 SLP 学习最早对 PLP 进行结构学习的工作之一，它首先收集训练数据中所有出现过的具体逻辑事实，然后根据它们的赋值分为

正/负样例，使用 Progol[109] 学习一个一阶逻辑规则集，最后再使用 MLE 学习其参数。De Raedt 等人[35] 提出的 PLP 模型压缩通过贪心算法逐条删除规则，并保证每次删除规则后得到的 PLP 关于训练样本的似然最大。该方法虽然能够对 PLP 模型结构进行修改，但它无法从数据中学习新规则。Bellodi 和 Riguzzi 提出的 SLIPCASE （Structure LearnIng of ProbabilistiC logic progrAmS with Em over bdds）[6] 和 SLIPCOVER（Structure LearnIng of Probabilistic logic programs by searChing OVER the clause space）[8] 分别通过 ILP 中的规则精化算子[121] 和演化计算以生成候选模型结构集，并通过 EMBLEM 算法[7] 学习模型参数。

基于 PILP 的知识精化的复杂度较高，因为它与 SRL 的结构学习相似，需要同时学习一阶逻辑规则结构学习和概率分布参数。不过，PILP 能在一定程度上利用一阶逻辑演算的特性对概率推断进行加速，因此相对 SRL 的模型学习来说更加高效。此外，PILP 的输入仍然是由逻辑事实构成的数据，因此它和 ILP 一样只能被应用在此类学习问题中进行知识精化。

1.3 有待研究的问题

如上节中所言，目前将一阶逻辑领域知识与机器学习相结合的方法大多均局限于小部分已有逻辑符号表示的领域。常规的机器学习方法仍然难以对一阶逻辑表示的领域知识加以利用，且以往的知识精化方法难以从一般领域中学习一阶逻辑规则。因此，如何更有效地将逻辑领域知识与机器学习相结合还有很多问题需要解决：

1. 如何在常规的机器学习方法中引入一阶逻辑表示的领域知识，以提高机器学习性能。在许多实际任务中均存在着关系型领域知识，常规的机器学习方法，例如统计学习与深度学习，难以对此类领域知识加以利用。这些领域知识均能够方便地通过一阶逻辑语言表示，但由于它们结构复杂，很难被应用在机器学习中。
2. 如何通过机器学习模型对领域知识精化进行改进。传统基于逻辑表示的知识精化方法，例如 ILP 和 PILP，只能在由逻辑事实表示的数据中实现领域知识精化。然而，大部分实际任务中的数据却不一定拥有逻辑符号表示。另一方面，常规的机器学习技术能够很方便地从此类数据中进行学习，但它们难以做到知识精化。

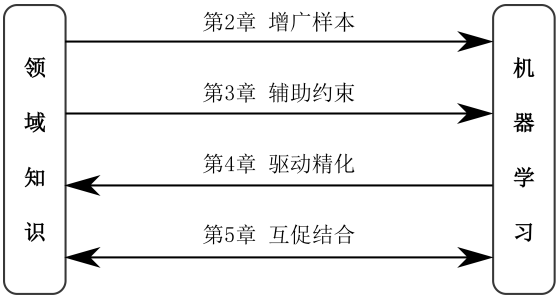


图 1.2 本文主要工作

3. 如何在学习过程中结合机器学习和领域知识，令二者相互促进、共同提升。将数据驱动的机器学习与知识驱动的逻辑推理相结合是人工智能领域中的核心挑战之一。机器学习方法能够学习如何将数据转化为类别和标记等符号，逻辑推理则能够基于领域知识进行复杂推理和知识精化。以往对二者的研究大部分是分离的，如何使它们互促结合仍然有待探索。

1.4 本文工作

本文对如何将一阶逻辑领域知识与机器学习相结合进行了研究，针对上一节中提出的三个问题，主要完成四项工作，分别对应于本文的第2至5章，图1.2直观地展示了这些工作之间的关系。

在第2章中，针对常规机器学习方法难以对关系型领域知识进行利用这一问题，本文提出了一种领域知识增广样本的机器学习方法——SUL（Statistical Unfolded Logic）学习，它能够将领域知识转化为样本，以提高机器学习性能。具体来说，SUL 学习可以通过领域知识构建新的特征，对数据的特征空间进行增广，以使得训练样本中包含更多与领域知识相关的信息。在 SRL 学习任务中的实验结果显示，与其他未引入领域知识的 SRL 方法相比，SUL 学习能够取得更好的效果。

在第3章中，针对一阶逻辑领域知识难以被应用在机器学习优化过程中这一问题，本文提出了一种领域知识辅助约束的机器学习方法——LASIN（Logical Abduction and Statistical INduction），它可以将领域知识转化为对机器学习模型搜索空间的约束。具体来说，LASIN 通过这些约束来要求机器学习在优化过程中所找到的模型与领域知识保持一致。本章将 LASIN 方法应用在了图像特征学习任务中，实验结果显示，引入一阶逻辑领域知识的 LASIN 所学得的特征比其他方法拥有更好的泛化性能。

在第4章中，针对一般数据中难以进行领域知识精化这一问题，本文提出了一种机器学习驱动的领域知识精化方法——KRL（Knowledge Refinement by Learning），它可以将领域知识精化问题转化为机器学习问题。具体来说，KRL 首先使用机器学习模型将数据转化为具体逻辑事实，以对领域知识进行增广，然后再使用基于一阶逻辑表示的机器学习方法从增广的领域知识中进行知识精化。实验结果表明，KRL 能够从图像中将领域知识精化为关于目标概念的形式化定义，且这些知识能够被直接重用于其他知识精化与推理任务。

在第5章中，针对如何在学习过程同时精化领域知识与提升机器学习性能的问题，本文提出了一种领域知识与机器学习互促结合框架——反绎学习（Abductive Learning），它通过机器学习将数据转化为增广领域知识的逻辑事实，以用于进行知识推理与精化；并将领域知识推理所得的逻辑事实作为训练机器学习模型的监督信息，以提升机器学习性能。基于此框架，本文还提出一种将深度学习与反绎逻辑程序相结合的 NLM（Neural Logical Machine）方法，并在一类复杂的机器学习任务中验证了反绎学习的有效性。

第二章　一种领域知识增广样本的机器学习方法

2.1　引言

在许多机器学习任务中，领域内存在大量的关系型知识，它们可以通过描述对象间的关系表示领域中隐含的结构信息。事实上，这一类领域知识往往可使用一阶逻辑语言表达。而常规的机器学习方法，例如深度学习[55]和统计学习[62]，由于表达能力的限制很难对此类知识加以利用。

统计关系学习（Statistical Relational Learning，SRL）[54]是一类可以对关系型知识进行利用的机器学习方法。它先将领域中的对象及它们之间的关系信息转化为随机变量以及条件依赖关系，并使用概率图模型（Probabilistic Graphical Models，PGM）[83]对领域进行建模，最终通过统计学方法从训练数据中进行学习。

与常规的机器学习方法相比，SRL 的学习效率较低，且学习性能易受领域知识质量的影响。SRL 所使用的模型既包括了含有关系信息的模型结构，又包含了描述领域中概率分布的模型参数。因此，SRL 的学习过程一般需要对这模型的这两个部分进行交替地优化（见第1章中的图1.1）。而当表达关系信息的领域知识不够准确时，很容易导致学得的模型结构不可靠，进而影响参数学习的效果[83]。

本章提出的 SUL 学习（Statistical Unfolded Logical Learning）是一种利用领域知识增广样本的机器学习方法，其目标是将关系型领域知识转化为更多的训练样本以提高机器学习的性能。具体来说，SUL 学习可以分为以下 3 个步骤：1) 将关系型领域知识转化为一组新特征，并使用这些特征对训练样本进行特征提取和增广；2) 在增广后的训练数据中使用效率较高、性能较为鲁棒的统计机器学习方法获得模型；3) 将统计学习模型转化为具有良好可理解性的一阶逻辑规则。实验显示，与其他未显式引入知识的学习的方法相比，经过领域知识增广样本的 SUL 学习能够取得更好的性能，并且拥有更高的学习效率。此外，被

转化为一阶逻辑规则后的模型能够较好地描述领域中的关系信息，具有良好的可理解性，且 SUL 学习能够在训练过程中实现一阶逻辑谓词发明[88]。

2.2 SUL 学习

SUL 学习的目标是在机器学习中引入关系型领域知识提高学习的性能。简单来说，SUL 先使用领域知识增广样本，再对增广后的训练数据使用机器学习得到模型。本节将首先对引入关系型领域知识的机器学习问题进行定义，然后再介绍 SUL 学习中使用领域知识增广样本的方法。

2.2.1 问题定义

SUL 主要面对的是监督学习问题。通常来说，监督学习任务的目标可使用一阶逻辑语言描述为如下形式：

$$B \cup H \models E. \tag{2.1}$$

其中的“$\models$”为语义蕴涵符，B 和 E 分别为输入的背景知识（background knowledge）和训练样本（examples），H 则为待学习的假设模型（hypothesis）。一般情况下，训练样本 E 分别由正样本 E^+ 与负样本 E^- 组成。式2.1表示监督的学习目标是从背景知识和样本中学得一个假设模型，以使得该模型与背景知识一起可以满足所有训练样本。

本文将领域知识库（domain knowledge base）记为 KB。如第2.1节中所介绍，领域中对象间的关系信息往往可以通过一阶逻辑规则表示。因此，本章中的 KB 为形如式1.1的一阶逻辑规则所组成的集合。

对于 ILP[107] 问题，式2.1中的 B、H 和 E 均为一阶逻辑规则集。因此，只需将 KB 加入背景知识即可将式2.1中的机器学习任务目标重写为：

$$B' \cup H \models E. \tag{2.2}$$

其中 $B' = B \cup KB$。

当领域中存在噪声时，可满足关系“$\models$”便很难完全成立。因此，PILP[34] 和 SRL[54] 选择将式2.2中的学习目标转化为最大化假设模型在训练数据中的后验概率似然（likelihood）$P(E \mid B', H)$。假设训练样本服从独立同分布，则有

$P(E \mid B', H) = \prod_{e \in E} P(e \mid B', H)$。我们通常使用对数似然（log-likelihood）函数将其中的连乘转化为求和：

$$LL_{B'}(H) = \log P(E \mid B', H) = \sum_{e \in E} \log P(e \mid B', H), \tag{2.3}$$

此时式2.2的学习目标可重写为：

$$\max_{H \in \mathcal{H}} LL_{B'}(H). \tag{2.4}$$

其中 $\mathcal{H}$ 为 H 的假设空间（hypothesis space）。

然而，由于 ILP、PILP 和 SRL 的假设模型 H 均为（赋权）一阶逻辑公式集合，此类模型的假设空间 $\mathcal{H}$ 十分复杂[22,83]，因此这些方法的学习效率往往不高。

对于常规的机器学习方法，如统计机器学习和深度学习，背景知识 B 为训练数据集中所有的示例 $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$，其中每个示例被表示为一个特征向量（feature vector）$\boldsymbol{x}_i \in \mathcal{X}$，而 $\mathcal{X}$ 为特征空间；训练样本 E 是数据集中每个示例的标记（label）$\{y_1, y_2, \ldots, y_m\}$，其中 $y_i \in \mathcal{Y}$，而 $\mathcal{Y}$ 为标记空间；假设模型 H 是一个用来预测示例标记的函数 $h : \mathcal{X} \mapsto \mathcal{Y}$。此时，式2.1中的学习目标可表示为经验风险最小化 (Empirical Risk Minimization，ERM) 形式[180]：

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \hat{R}(h), \tag{2.5}$$

这里的 $\hat{h}$ 为优化式2.5中目标所得到的假设模型，$\hat{R}$ 为 h 在训练数据 $\{\langle \boldsymbol{x}_i, y_i \rangle\}$ 中的经验风险（empirical risk）：

$$\hat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} L(h(\boldsymbol{x}_i), y_i), \tag{2.6}$$

其中 $L(h(\boldsymbol{x}_i), y_i)$ 为假设模型 h 在训练样本 $\langle \boldsymbol{x}_i, y_i \rangle$ 上的损失（loss）。

与 ILP、PILP 和 SRL 等方法相比，当损失函数 L 和假设模型 h 均拥有较好的性质时，式2.5中的 ERM 问题可以使用凸优化和反向传播等方法求解，且拥有较高的学习效率。不过由于表达能力的限制，它很难像式2.2与式2.4那样在学习过程中引入一阶逻辑规则表达的领域知识 KB。

为了解决此问题，本章提出用领域知识增广式2.6中的训练样本 $\{\langle \boldsymbol{x}_1, y_1 \rangle, \langle \boldsymbol{x}_2, y_2 \rangle, \ldots\}$，以增加学习中可以利用的信息，下一小节将详细介绍该方法。

2.2.2 领域知识增广样本

SUL 学习对样本进行增广的方式有两种：1) 将领域知识转化为特征（feature）作为原特征空间 $\mathcal{X}$ 的补充，以增加每个示例所包含的信息；2) 根据知识从领域中抽取更多的样本，以增加训练数据的大小。

具体来说，SUL 学习对样本的扩充方式是使用一阶逻辑中的演绎推理将领域知识 KB 转化为一组机器学习中可直接利用的特征 F。本章将 F 所张成的特征空间记为 $\mathcal{F}$，它会与 B 中的原特征空间 $\mathcal{X}$ 一起作为机器学习的输入空间并用于构建假设模型。因此，该过程可直观地理解为利用 KB 将机器学习的假设空间 *展开*（unfold）并增广为 $\mathcal{X} \times \mathcal{F}$。

此展开过程可以通过一阶逻辑中的*前向链*（forward chaining）推理完成。前向链是一种基于*三段论*（modus ponens）[43] 的演绎推理方式，常被使用于专家系统[71] 中。它指已知一阶逻辑公式 $\Gamma \to T$ 成立且公式中的前提 Γ 成立时可推出结论 T 成立。三段论的推理过程一般被表示为 $\{\Gamma, \Gamma \to T\} \vdash T$，其中“$\vdash$”为一阶逻辑中表示推出关系的语构蕴涵符。因此，SUL 学习的展开过程可使用相同的方法表达为如下形式：

$$B \cup KB \vdash F. \tag{2.7}$$

其中 $F = \{f_1, \ldots, f_k\}$ 为被展开的特征，B 是展开过程中可能使用的背景知识。

得到新特征后，便可将式2.2里 B' 中的 KB 替换为 F，将学习目标转化为：

$$B \cup F \cup H \vdash E. \tag{2.8}$$

即从 $B \cup F$ 构成的新假设空间 $\mathcal{H}'$ 中对模型 H 进行搜索，使得 H 与 $B \cup F$ 一起可以推出训练样本 E 成立。

需要注意的是，式2.8所使用的推出符“$\vdash$”与式2.2中使用的语义蕴涵符“$\models$”并不等价。这表示式2.8中问题的假设空间 $\mathcal{H}'$ 只是式2.2中问题假设空间 $\mathcal{H}$ 的子集。因为我们无法通过前向链推理获得所有满足式2.7的 f_j。事实上，当 KB 中存在形如

$$P \leftarrow Q_1 \wedge Q_2 \wedge \ldots \wedge P \wedge \ldots \wedge Q_s. \tag{2.9}$$

的递归规则时，可能存在无穷个满足式2.7的 f_j，但是在实际操作中，我们获得的 F 大小往往有限。

式2.8中的目标看起来与式2.2非常相似。但事实上，式2.2中的 KB 是一组难以被机器学习方法利用的一阶逻辑公式；而式2.8中的 $F = \{f_1, \ldots, f_k\}$ 已是一组可直接用于增广样本的特征。

基于以上方法所得到的 F，SUL 学习可采用 ILP[107] 中的命题化[90] 方法将其用于增广训练样本的特征。对于数据集 $B \cup E$ 中的每一个训练样本 $\langle \boldsymbol{x}_i, y_i \rangle$，SUL 学习会对其提取一组新的特征 $\boldsymbol{u}_i = \langle u_{i1}, u_{i2}, \ldots, u_{ik} \rangle$，其中：

$$u_{ij} = \begin{cases} 1, & f_j(\boldsymbol{x}_i) = true, \\ 0, & f_j(\boldsymbol{x}_i) = false. \end{cases} \tag{2.10}$$

最终将训练样本增广为 $\{\langle (\boldsymbol{x}_1, \boldsymbol{u}_1), y_1 \rangle, \ldots, \langle (\boldsymbol{x}_m, \boldsymbol{u}_m), y_m \rangle\}$，其中 $(\boldsymbol{x}_i, \boldsymbol{u}_i) \in \mathcal{X} \times \mathcal{F}$ 表示将 $\boldsymbol{x}_i$ 和 $\boldsymbol{u}_i$ 进行拼接后得到的增广特征向量。

如本小节开头所言，除了对训练样本的特征空间进行增广，SUL 学习还可以对训练样本的个数进行扩充。例如，ILP 和 SRL 任务往往采用闭世界假设（closed world assumption，CWA），该假设认为所有未出现在 E^+ 中的目标概念样本均为负样本，它可以用一阶逻辑表达为领域知识 KB_{CWA}：

$$\forall e (e \notin E^+ \rightarrow e \in E^-). \tag{2.11}$$

这时，SUL 学习同样可以采用前向链推理将该领域知识增广为新的负样本：

$$B \cup KB_{CWA} \vdash \bar{E}^-. \tag{2.12}$$

然后，它可以将 $E \cup \bar{E}^-$ 共同作为增广后的训练样本 E' 用于 SUL 学习：

$$B \cup F \cup H \vdash E'. \tag{2.13}$$

当 KB 增广生成的候选特征数量非常多时，SUL 学习可以通过其他领域知识进一步限制 F 的大小和形式。例如，我们可以在展开过程可通过 ILP 中的精化算子（refinement operator）[121] 或模式声明（mode declaration）[109] 约束式2.7中的展开过程。这些精化算子与模式声明同样可使用一阶逻辑规则表示，因此它们均可以被直接引入 SUL 学习的展开过程。

此外，SUL 学习展开 KB 所得到特征 $f_i \in F$ 均包含一部分 KB 中的语义，所以它们拥有良好的可理解性，且能够以一阶逻辑规则的形式表达。因此，SUL 学习从增广特征空间 $\mathcal{X} \times \mathcal{F}$ 中学得的机器学习模型可以通过一定手段重新

转化为具有良好可理解性的一阶逻辑规则。例如，决策树从根节点到叶节点的路径即可直接作为一阶逻辑规则的结构，其中每个中间节点所使用的特征 f_i 即为该规则中的一阶逻辑文字。这种转化过程可看作将被展开的假设空间重新折叠（folding）为一阶逻辑公式。

2.2.3 SUL 学习中的谓词发明

事实上，从一阶逻辑归纳的角度来看，SUL 学习在使用 KB 增广样本特征空间时所得到的 F 是一组可用于描述领域内部结构的候选新谓词。当对增广的样本进行学习时，机器学习可以对这些候选新谓词进行筛选，并保留对分类精度贡献最高的特征作为最终输出的新发明谓词。

SUL 的样本特征增广和筛选是一种*构造性归纳*（constructive induction）[88] 过程。一般来说，构造性归纳的目标是在提升模型分类的精度的同时降低其复杂度[172]。不同于那些从已有特征空间中进行特征学习的方法，构造性归纳往往需要借助先验知识主动生成特征。对于 SUL 学习来说，KB 即是在构造性归纳过程中使用的先验知识。

在以逻辑为表达的机器学习中，谓词发明是一个组合优化问题。对于内部结构较复杂的领域，SUL 学习方法通过2.7展开所得的候选新谓词数量通常十分庞大，而其中绝大部分可能并无用处。但是，借助机器学习，SUL 可以做到十分高效的谓词筛选，最终发明出真正对描述领域结构有帮助的新谓词。

2.3 实验验证

在本节中，我们将介绍 SUL 方法在 SRL[54] 任务中的应用，并提出一种基于关系型路径（relational path）[139] 领域知识的机器学习算法 SUL-Path。随后，本节将在 5 个真实 SRL 任务上比较 SUL-Path 方法与其他未引入领域知识的 SRL 方法的性能。

2.3.1 SUL 在统计关系学习中的应用

本小节提出的 SUL-Path 算法是 SUL 学习的一种实现，它使用关于关系型路径（relational path）[139] 的领域知识增广训练样本，下面将对 SUL-Path 方法进行简单介绍。

SRL[54] 的学习目标如2.4式所示，其中的 H 为赋权一阶逻辑规则（weighted first-order logical rule）构成的集合。赋权一阶逻辑规则的定义与第1章中的式1.3一致。为避免符号混淆，本章中将其表示为：

$$w :: T \leftarrow S_1 \wedge S_2 \wedge \ldots \wedge S_s. \tag{2.14}$$

其中 w 为规则的权重，表示规则为 *true* 的可能性；S_i 和 T 均为一阶逻辑文字，T 为关于 n 元谓词目标概念（target concept）t/n 的原子公式 $t(X_1,\ldots,X_n)$，S_i 为背景知识 B 中其他谓词构成的文字。SRL 的目标是从训练数据 $B \cup E$ 中学习赋权一阶逻辑规则构成的模型假设 H 以描述目标概念（target concept）t/n，并对未见原子 $t(X_1,\ldots,X_n)$ 辅助为 *true* 的概率进行预测。

本节提出的 SUL-Path 采用了 Richards 和 Mooney[139] 提出的关系型路径假设（relational path assumption）作为领域知识，我们将其记为 KB_{path}。该假设认为，对于所有作为参数同时出现在正样本事实中的逻辑常量，在 $B \cup E$ 构成的**关系型超图**（relational hypergraph）中总存在长度有限的路径连接它们。而这些长度有限的路径即可作为特征 F 以增广训练样本。它们可表达为：

$$\forall i, j \in [1..l](t(X_1,\ldots,X_l) \leftarrow i \neq j \wedge path(X_i, X_j)). \tag{2.15}$$

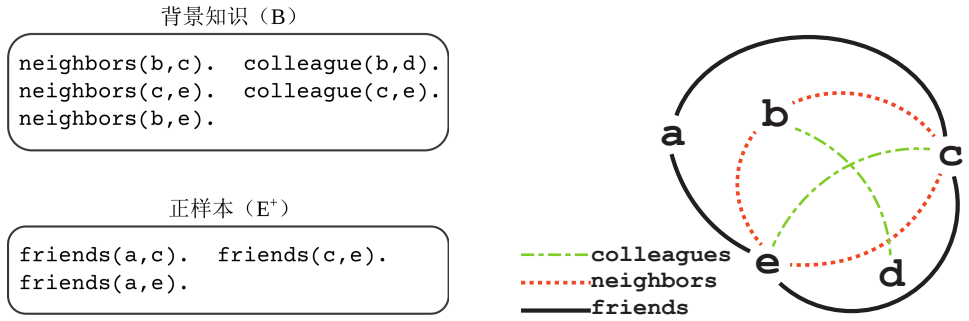
$$\exists r(path(X,Y) \leftarrow r(X,Z) \wedge path(Z,Y)). \tag{2.16}$$

其中规则2.15表示对于任意符合目标概念的逻辑事实 $t(X_1,\ldots,X_n)$，它的两个不同的参数 X_i 与 X_j 之间一定存在一条“路径” $path(X_i,X_j)$ 连接它们。规则2.16则对“路径”进行了递归定义，它表示若 B 中存在一个事实 $r(X,Z)$，且 Z 至 Y 存在一条路径，则 X 至 Y 存在一条路径。

为了更好地理解上面定义的领域知识 KB_{path}，我们首先定义将背景知识 B 和训练样本 E 构成的**关系型超图**：

定义 2.1 (关系型超图) 给定背景知识 B 和训练样本 E，可以生成一个**关系型超图** $\mathcal{G}(B \cup E) = (V, L)$，其中的每个点 $v \in V$ 是 $B \cup E$ 中的一个逻辑常量，每条**超边**（hyperedge）$l \in L$ 是 $B \cup E$ 中的一个具体原子公式。

图2.1展示了一个从背景知识 B 和训练样本 E 构建关系型超图的例子。其中图2.1(a)分别展示了领域中关于目标概念 $friends/2$ 的背景知识和训练样本，它们均由具体事实构成，而每条具体事实在图2.1(b)中构成一条超边。



(a) 背景知识和正样本 **(b)** 关系型超图

图 2.1 关于目标概念 $friends/2$ 的关系型超图

可以看到，对于图2.1(a)中正样本 $friends(c,e)$ 的参数 c 和 e，存在多条路径联通它们，例如：$[colleague(c,e)]$ 和 $[neighbors(b,c), neighbors(b,e)]$。根据领域知识 KB_{path}，它们均满足2.16式中定义的 $path$ 概念。显然，KB_{path} 的直观含义是任意目标概念总可以被定义为有限个构成 $path$ 的逻辑文字的合取。在实际应用中，这种假设十分合理。

根据 KB_{path}，本文定义 SUL-Path 用于增广样本的第一种特征——路径特征（path feature）如下：

定义 2.2 (路径特征) 给定一个关系型超图 $\mathcal{G} = (V, L)$ 以及关于目标概念 t/n 的正样本 $e^+ = t(v_1, \ldots, v_n)$，由 e^+ 和 $\mathcal{G}$ 生成的具体路径是一个连通 $\{v_1, \ldots, v_n\}$ 的超边序列 $p = [l_1, \ldots, l_k]$。路径特征则是 p 的泛化形式，即将 p 中所有不同逻辑常量分别替换为不同逻辑变量后的一阶逻辑原子序列。

例如，考虑图2.1中的超图和正样本 $e^+ = friends(c,e)$，由它们生成的一条具体路径为如下形式：

$$p = [neighbors(b,c), neighbors(b,e)], \tag{2.17}$$

它可以被泛化为一个关于目标概念 $friends(X,Y)$ 的路径特征：

$$f = [neighbors(Z,X), neighbors(Z,Y)] \tag{2.18}$$

在路径特征的基础上，SUL-Path 还定义了局部路径特征（local path feature）和分支特征（branch feature）以增广训练样本，它们的严格定义分别如下：

定义 2.3 (局部路径特征) 给定一条具体路径 $p = [l_1, \ldots, l_k]$，若它经过的所有节点为 $V_p = \{v_1, \ldots, v_s\}$，则对任意 $V'_p \subseteq V_p$，连通 V'_p 的路径特征被称为 p 的*局部路径特征*。

定义 2.4 (分支特征) 对于 $\mathcal{G} = (V, L)$ 中的节点集合 $V_1 \subset V$，路径 $p = [l_1, \ldots, l_k]$ 被称为 V_1 的*分支*当且仅当：

1. $|V_1 \cap V_p| = 1$，其中 V_p 为 p 所经过的节点；
2. V_1 与 V_p 的交点 v 是 p 的端点，即 v 只出现在 l_1 或 l_k 中。

p 泛化后得到的路径特征则被称为 V_1 的*分支特征*。

其中，局部路径特征可以描述路径特征中某些节点之间的局部关系，它在一些关系较复杂、关系型路径较长的领域中十分有意义。例如在2.3.2小节实验中的 Cora 数据集里，定义“相同引用”关系的目标概念 *samebib*/2 有如下这一条路径特征：

$$f(X_1, X_2, X_3, X_4, X_5) = [title(X_1, X_3), haswordtitle(X_3, X_4), haswordtitle(X_5, X_4), title(X_2, X_5)] \tag{2.19}$$

其中的 X_i 均为具体路径在泛化时用于替换常量的逻辑变量。该特征表示两篇论文 X_1 和 X_2 的标题分别为 X_3 和 X_5，它们中同时出现了一个单词 X_4。显然，这种情况下 X_1 和 X_2 为同一篇文章的概率并不高，除非它们的标题中还含有更多其它的相同单词。这时，SUL-Path 便通过连接 X_3 和 X_5 的局部路径特征 $[haswordtitle(X_3, Y), haswordtitle(X_5, Y)]$ 对式2.19中的路径特征进一步增广。

分支特征在超图中的形状类似于从路径中伸出去的树枝，它们可以描述路径中某些节点自身的属性。例如对于图2.1中的领域，若背景知识中存在着像 $male(X)$ 或 $female(X)$ 用于描述对象属性的一元谓词，它们可能对学习目标概念 $friends/2$ 同样有一定帮助。

每个路径特征都可以表示为一组一阶逻辑原子公式的合取，例如2.19中的路径特征即可表示为：

$$f(X_1, X_2, X_3, X_4, X_5) \leftarrow title(X_1, X_3) \wedge haswordtitle(X_3, X_4) \wedge haswordtitle(X_5, X_4) \wedge title(X_2, X_5) \tag{2.20}$$

不难证明，路径特征、局部路径特征和分支特征构成的合取式足以表达任意形

如 2.14式的带权逻辑规则结构。但另一方面，这些特征的个数会随着路径长度的增长而指数级的增加。根据 Richards 和 Mooney[139] 提出的关系型路径假设，最优规则集 R^* 只会存在于由长度较短的路径特征构成的假设空间中。因此，SUL-Path 并不需要生成长度太大的路径特征，下面的实验也佐证了这一观点。

简而言之，SUL-Path 在使用 KB_{path} 增广训练样本时，会首先构造一个超图，并在图中对正样本进行超路径查找和泛化以获得特征 F。在构造属性-值数据集 D 时，SUL-Path 会将每个 $f_j \in F$ 转化为形如2.20式的逻辑公式，然后检测每个样本 $e_i \in E$ 是否满足 f，并以之对 e_j 的特征空间进行增广。然后，SUL-Path 会采用决策树模型[180] 在 D 中进行机器学习。最终在折叠阶段将决策树模型重新转化为赋权逻辑规则[117]。

2.3.2 实验设置

实验数据

本节的实验中共使用了两个 SRL 真实数据集，它们分别为：

- **Cora**：该数据集收集了一部分计算机科学领域中的论文引用条目，其目标是为了找到数据中重复的引用[134]。该数据的背景知识中拥有 5 类共 3079 个逻辑常量，10 种谓词以及 687422 个具体事实。其中的原始概念为 $author(Bib, Author)$、$title(Bib, Title)$、$venue(Bib, Venue)$ 等等，它们描述了引用条目中每个部分的内容。该任务的目标概念分别是 *sameauthor*/2、*samebib*/2、*sametitle*/2 和 *samevenue*/2，用于预测两条不同的引用内容是否表示同一对象。为简便起见，下文将这四个任务分别记为 *author*、*bib*、*title* 和 *venue*。
- **UW-CSE**：该数据集合收集了华盛顿大学计算机系成员间的关系[36]。它的背景知识中包含了 9 类共 929 个常量，12 种谓词以及 260254 个具体事实。其中的原始概念定义了每个对象的身份（学生或教授）、授课关系和论文发表情况等等。该数据集中的目标概念为 *advisedby*/2，用来判断某个学生是否由某个教授指导。下文将该任务记为 *advisedby*。

在这两个数据集中，UW-CSE 的数据量虽然较少，但领域结构更复杂。因此虽然 Cora 数据集虽然拥有更多常量，但它的任务比 UW-CSE 更简单。这两个数据集均满足式2.11中的 CWA 假设，因此 SUL-Path 在对样本特征空间进行增广的同时还能对样本数量进行增广。

对比方法

在本节的 SRL 任务中，我们一共比较了 4 种方法，它们分别是：

1. **RDN-Boost**[116]：该方法将带权逻辑规则学习转化为一个概率图模型学习问题。它采用伪似然（pseudo-likelihood）[83] 函数对式2.4的目标进行近似，并采用 FGB（Functional Gradient Boosting）方法对规则的结构和权重进行优化。在本节的实验中，我们采用了原论文中的参数[116] 进行训练。此外，为了缩小规则结构的搜索空间，RDB-Boost 需要在背景知识中预先定义一组“模式”（mode）作为规则结构的基本组成部分。实验中我们枚举了所有可能的模式以保证 RDN-Boost 的完备性，在部分实验上甚至取得了比原文更好的结果。由于 RDN-Boost 在训练过程中需要使用大量负样本，我们在实验中采用了两种不同的设置：RDN-Boost-org 仅仅使用了训练集中的负样本；RDN-Boost-total 则补全了所有训练集中未出现的负样本作为增广。
2. **Alchemy**[36]：这是目前为止应用最广泛的统计关系学习工具包之一，它采用的 LSM 算法[82] 可以交替地学习马尔可夫逻辑网的结构和参数，并使用伪似然函数对原学习问题进行放松。本节的实验中，我们使用的是其最新版本 Alchemy 2.0，并采用工具包中默认的算法参数进行试验。
3. **ALEPH++-MLN**[66]：该算法采用判别式学习进行学习。它首先使用 ILP 工具包 ALEPH[①]生成一系列候选逻辑规则，并将它们作为马尔可夫逻辑网的初始结构，然后再使用梯度下降学习各个规则的权重并剔除多余的规则。ALEPH 同样需要通过设置模板以缩小规则结构的搜索空间，我们采取与 RDN-Boost 同样的方法设置这些模板。此外，我们按照原文的建议设置该方法的参数。
4. **SUL-Path**：这是本节中提出的算法。对于 *author*、*bib*、*title*、*venue* 和 *advisedby* 五个任务，SUL-Path 在使用 KB_{path} 增广样本时所使用的最大关系型路径长度分别为 4、4、4、4 和 2。在机器学习阶段，我们测试了多种基于决策树的机器学习模型，包括 J48、AdaboostM1 和随机森林，下文中使用这些机器学习模型的 SUL-Path 方法分别被记为 SUL-Path-J48、SUL-Path-Boost 和 SUL-Path-RF。SUL-Path 在增广样本时能够自动对训练样本中未出现的大量负样本进行采样，因此在进行机器学习时中会出现类别不平衡的情况。在实验中我们采用了 Weka[61] 的 CostsensitiveClassifier 进行训练，其中错分正样本的代价被设置为负样本的两倍。

[①] http://www.cs.ox.ac.uk/activities/machlearn/Aleph/aleph.html

表 2.1 对比方法在每个任务中的 AUC 值

算法	*author*	*bib*	*title*	*venue*	*advisedby*
SUL-Path-J48	0.994 ± 0.004	0.926 ± 0.025	0.921 ± 0.031	0.839 ± 0.046	0.633 ± 0.071
SUL-Path-Boost	**0.998 ± 0.002**	**0.998 ± 0.001**	**0.987 ± 0.011**	**0.977 ± 0.011**	0.975 ± 0.015
SUL-Path-RF	**0.998 ± 0.002**	0.989 ± 0.009	0.982 ± 0.017	0.946 ± 0.023	**0.992 ± 0.006**
RDN-Boost-org	0.985 ± 0.014	0.916 ± 0.021	0.706 ± 0.121	0.589 ± 0.040	-
RDN-Boost-total	0.986 ± 0.012	0.949 ± 0.016	0.729 ± 0.126	0.590 ± 0.038	0.983 ± 0.014
Alchemy	0.597 ± 0.152	-	0.604 ± 0.216	-	0.393 ± 0.103
ALEPH++-MLN	-	-	-	-	0.127 ± 0.032

为了验证学得的赋权逻辑规则是否能够真正描述领域中的全局概率分布，每个任务的测试样例均包含了目标概念所有可能的具体逻辑事实，这造成测试数据中的负样本数量远远大于正样本。因此，本节的实验中采用 AUC 指标衡量所有对比方法的表现。此外，为了衡量这些方法表现是否稳定，所有任务均采用 5 折交叉验证的方式进行练和测试。

2.3.3 实验结果

表2.1展示了所有算法在 5 个任务中的结果。由于 UW-CSE 数据集中未提供负样本，所以 RDN-Boost-org 在其中没有结果。在 *bib* 和 *venue* 任务中，Alchemy 在 100 小时内未能完成学习；ALEPH 在所有 Cora 的任务中均由于内存容量不足而失败，所以 ALEPH++-MLN 在这些任务中均未取得结果。

可以看到，SUL-Path 在所有任务中均取得了最佳结果。ALEPH++-MLN 的表现最差，这是因为它所使用的归纳逻辑程序设计方法 ALEPH 在学习规则结构时无法应对数据中的噪声[80]。在所有 SUL-Path 方法中，SUL-Path-J48 表现最差。这是因为这些任务中的目标概念均十分复杂，需要许多条规则才能完整描述。而 SUL-Path-J48 只能学习一棵决策树，所以会漏掉许多有意义的规则。对于最复杂的任务 *advisedby* 来说，该缺陷对算法表现的影响尤为严重。

表2.2中记录了对比算法的训练时间。所有实验均在一台 12 核 2.53GHz 的 CPU 集群上完成。可以看到，基于判别式学习（discrimitative learning）的 SUL-Path 和 ALEPH++-MLN 的训练速度远高于其他生成式模型（generative model）学习方法。因为 ALEPH 只从数据中学习长度为 4-8 的非递归一阶逻辑规则，所以 ALEPH++-MLN 运行速度最快。

然而，相比于学习阶段的速度，SUL-Path 在展开假设空间时消耗的时间较长。表 2.3分阶段展示了每个任务中的展开时间以及新生成的属性-值数据

表 2.2 对比算法的平均训练时间（秒）

算法	*author*	*bib*	*title*	*venue*	*advisedby*
SUL-J48	0.2	87.2	0.5	1.5	0.4
SUL-Boost	1.1	302.6	2.2	9.5	1.1
SUL-RF	1.6	471.0	4.6	19.7	1.6
RDN-Boost	25.6	947.8	278.6	204.1	15.6
Alchemy	5232.3	-	5898.1	-	5478.0
ALEPH++-MLN	-	-	-	-	0.07

表 2.3 SUL-Path 每个展开阶段消耗的时间（秒）与增广样本的大小

	时间开销		增广样本大小	
任务	知识转化特征	计算特征向量	特征维度	样本数量
author	5.7	13.9	36	2428
bib	47.0	1475.5	64	457782
title	20.6	80.8	34	16396
venue	55.4	491.6	33	73354
advisedby	11.5	22.7	452	10625

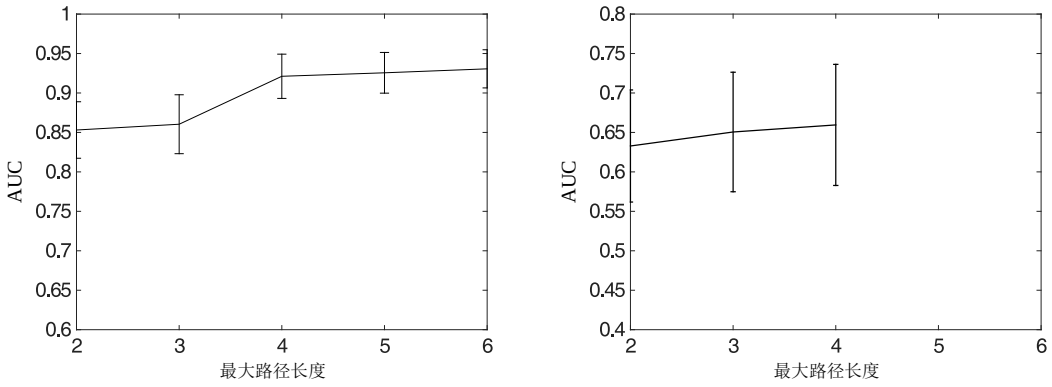
集大小。若将展开阶段的时间考虑进来，SUL-Path 的训练时间甚至超过了 RDN-Boost。不过它的时间消耗可以通过改进关系型路径查找的实现而减少，例如采用并行计算等等。

本章实验中我们还研究了最大关系型路径长度对 SUL-Path 表现的影响。由于 SUL-Path-Boost 和 SUL-Path-RF 的结果已经接近最优，我们选择通过观察 SUL-Path-J48 的结果进研究，该结果如图2.2所示。由于领域内的关系过于复杂，*advisedby* 任务中当最大路径长度超过 4 时，SUL-Path 的展开阶段就无法在 1 小时内结束，因此我们没有计算它的结果。

可以观察到，增加路径长度确实对提升 SUL-Path 的表现有所帮助。不过一旦长度超过某个阈值，它带来提升就变得十分有限了。这表示 SUL-Path 并不需要太长的关系型路径即可学得接近最优的假设模型，该结果一定程度上验证了 Richards 和 Mooney[139] 提出的关系型路径假设。

2.3.4 SUL-Path 的谓词发明

SUL-Path 的输出是一个标准 Prolog 引擎[14] 可直接运行的 Prolog 规则集合，它同时包含了被发明的谓词以及由折叠而得的一阶逻辑规则。与一般的 SRL 模型相比，这种形式的模型拥有更好的可理解性。例如，在 *author* 任务



(a) *title* **(b)** *advisedby*

图 2.2 最大关系型路径长度对 SUL-Path-J48 性能的影响

中，SUL-Path-Boost 输出的一条一阶逻辑规则即为：

$$\begin{aligned} tree_0_score(X_1, X_2, W) \leftarrow & ct_path_0_0(X_1, X_2, N) \\ & \wedge N > 0.0 \wedge W = 1.0. \end{aligned} \tag{2.21}$$

此规则中的谓词 $tree_0_score/2$ 是目标概念 $sameauthor/2$ 的赋权形式；最后一项 $W = 1.0$ 表示该规则的权重；谓词 ct_path_* 能够计算背景知识中路径特征 $path_*$ 被满足的次数；$path_0_0$ 即为被发明的谓词，它的编号表示他是路径 $path_0$ 加上它的一条局部路径。这两个被发明的谓词分别是：

$$\begin{aligned} path_0(X_1, X_2, X_3, X_4, X_5) \leftarrow & author(X_3, X_1) \wedge title(X_3, X_4) \\ & \wedge title(X_5, X_4) \wedge author(X_5, X_2). \\ path_0_0(X_1, X_2, X_3, X_4, X_5) \leftarrow & path_0(X_1, X_2, X_3, X_4, X_5) \\ & \wedge haswordauthor(X_1, X_6) \\ & \wedge haswordauthor(X_2, X_6). \end{aligned}$$

根据规则中其他谓词的意义可以解读出，这两个新谓词事实上表示如下意思：

1. $path_0$ 表示作者 X_1 和 X_2 分别撰写了两篇标题一样为 X_4 的论文 X_3 和 X_5；
2. $path_0_0$ 则表示若 X_1 和 X_2 满足 $path_0$，则他们的名字中至少还拥有一个相同的单词 X_6。

因此，规则2.21意味着只要 X_1 和 X_2 满足 $path_0_0$ 所描述的关系，他们 100%

是同一个作者。借助不同新谓词的组合，SUL-Path 可以学得拥有较大长度规则，因此能够表示关系较为复杂的目标概念，这在目前其他的 SRL 和 PILP 算法中较难做到。

2.4 小结

本章提出的 SUL 学习是一种利用一阶逻辑领域知识增广样本的机器学习方法。它可以将常规机器学习难以利用的关系型领域知识转化为样本以提升机器学习的效果。SUL 学习可以通过两种方式增广训练样本：第一种是将关系型领域知识转化为机器学习可以直接利用的特征，它可以对样本的特征空间进行增广，使得增广后的特征向量包含更多与领域知识相关的信息；第二种则是根据 CWA 假设对训练样本的数量进行增广，通过采样更多的训练样本来提升机器学习的效果。除了提升机器学习的性能，SUL 学习使用领域知识增广的样本特征还具有良好的可理解性，因此它能够以较高的效率实现 SRL 中的谓词发明。作为 SUL 学习的一类算法实现，SUL-Path 可以采用关系型路径假设作为领域知识来对 SRL 任务中的训练样本进行增广。实验验证，与未引入领域知识的 SRL 方法相比，SUL-Path 在实际任务中可以取得更优异的效果。

本章的主要工作：

- W.-Z. Dai and Z.-H. Zhou. Statistical unfolded logic learning. **In: Proceedings of the 7th Asian Conference on Machine Learning (ACML'15)**, Hong Kong, China, 2015, JMLR: W&CP 45, pp. 349-361. （机器学习领域重要国际会议）

第三章　一种领域知识辅助约束的机器学习方法

3.1　引言

本章考虑与第2章类似的设定，即通过引入一阶逻辑表示的领域知识促进机器学习的效果。不过与上一章不同的是，本章提出的方法试图将领域知识转化为机器学习优化过程中的辅助约束提升学习性能。

上一章中提出的 SUL 学习主要通过领域知识增广训练样本帮助机器学习。然而在某些任务中，我们拥有的领域知识可能不够强或不够准确，此时便无法通过式2.7 中的展开方法增广出质量足够好的训练样本。这种情况下，一次性将领域知识转化为训练样本的机器学习方法不一定能够取得最佳的效果。

事实上，人们在生活中不仅经常面对这样问题，并且还拥有一套成熟的解决办法。当碰到未遇见过的问题时，人们往往会利用自己的领域知识针对该问题猜想一个可能的解决方案。由于使用的领域知识不一定足够强，因此该解决方案不一定最优。所以在接下来验证该方案是否可以解决问题的过程中，我们会不断地从环境中收集新的信息，并根据领域知识对目前的解决方案进行更新，直至找到问题的最优解。

心理学家们将该办法称为主动假设检验策略（active hypothesis-testing）[136]，人们在观察未知事物、学习解决未知的数学问题时经常采用它。该策略在进行假设检验的过程中，不断地通过领域知识生成候选假设，然后使用环境中所收集的信息对假设进行优化和更新，直至收集所得的新信息均无法对该假设进行证伪时，它便被作为问题的解被接受。显然，在此过程中，领域知识被作为一种约束以限制解的搜索空间，而收集环境信息以在解空间中进行优化则可看作一种数据驱动的学习方法。

受以上过程的启发，本章提出了 LASIN（Logical Abduction and Statistical INduction）方法实现领域知识对机器学习的辅助约束。LASIN 能够使用反绎逻辑推理（logical abduction）[101,127] 以根据一阶逻辑表达的领域知识约束学习任

务的假设空间，然后采用统计归纳（statistical induction）在被约束过的假设空间中进行机器学习。这两个步骤同样可重复迭代地进行，直至找到符合要求的假设模型。

本章在手写字符图片的特征提取任务中对 LASIN 方法进行了实验。结果表明，与未使用领域知识的特征提取方法相比，LASIN 提取出的特征能够在分类和迁移学习任务中取得更好的效果。此外，实验结果还显示它提取的特征与领域知识基本吻合，因此也更符合人类的认知习惯。

3.2 LASIN 方法

LASIN 的基本思想是在机器学习的优化过程中将领域知识作为辅助约束。本节将首先对 LASIN 方法的形式化定义进行介绍，随后再详述该方法如何将领域知识转化为约束并应用于机器学习。

3.2.1 问题定义

LASIN 与第一章中的 SUL 方法学习目标类似。其输入包括一个由一阶逻辑公式集构成的领域知识库 KB 和一个训练集 $B \cup E$。其中背景知识 B 和样例 E 可被表示为一个训练集 $D = \{\langle \boldsymbol{x}_1, y_1\rangle, \ldots, \langle \boldsymbol{x}_m, y_m\rangle\}$，里面的每个样本分别由特征向量 $\boldsymbol{x}_i \in \mathcal{X}$ 与标记 $y_i \in \mathcal{Y}$ 构成，这里的 $\mathcal{X}$ 与 $\mathcal{Y}$ 分别表示问题的输入与输出空间。LASIN 目标是学习一个用于预测未见示例标记的函数 $h : \mathcal{X} \mapsto \mathcal{Y}$。

若不考虑领域知识 KB 对学习问题的约束，LASIN 的学习任务即退化为式2.5中的 ERM 问题。否则在一般情况下，LASIN 的目标可表示为如下形式：

$$\begin{aligned} \min_{h \in \mathcal{H}} \quad & \hat{R}(h) \\ \text{s.t.} \quad & \mathrm{Con}(KB \cup \{h\}). \end{aligned} \tag{3.1}$$

该目标函数下的约束表示学得的假设模型 h 必须与 KB 一致（consistent）[43]，即此二者不产生矛盾（$\nexists \varphi((\{h\} \cup KB \models \phi) \wedge (\{h\} \cup KB \models \neg\varphi))$）；而 $\hat{R}$ 依然表示 h 在训练数据 D 中的经验风险：

$$\hat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} L(h(\boldsymbol{x}_i), y_i). \tag{3.2}$$

其中 $L(h(\boldsymbol{x}_i), y_i)$ 为假设模型 h 在训练样本 $\langle \boldsymbol{x}_i, y_i\rangle$ 上的损失。

算法 3.1 LASIN

算法输入：训练数据 $D = \{\langle \boldsymbol{x}_1, y_1\rangle, \ldots, \langle \boldsymbol{x}_m, y_m\rangle\}$；一阶逻辑领域知识库 KB
算法过程：
$t = 0$;
$h_0 = \Phi$;
// 检验当前学得的假设模型 h_t 是否足够满足输出条件;
while $t < T$ 或 $|error_t| \geq threshold$ **do**
  // 从数据中收集关于 h_{t-1} 错误的信息;
  $error_t = \cup_{i=1}^{m} \text{difference}(h_{t-1}(\boldsymbol{x}_i), y_i)$;
  // 根据 $error_t$ 和 KB 进行反绎推理得到新的候选假设空间 $\mathcal{H}_t$;
  $\mathcal{H}_t = \text{abduce}(error_t, KB)$;
  // 在假设空间 $\mathcal{H}_t$ 中进行机器学习;
  $h_t = \arg\min_{h \in \mathcal{H}_t} \hat{R}(h)$;
  $t = t + 1$
end while
算法输出： h_t

对于常规的机器学习问题，$\mathcal{X}$ 一般为向量空间 $\mathbb{R}^d$。LASIN 的难点在于如何使用一阶逻辑表达的领域知识约束向量空间中的机器学习。由于一阶逻辑公式所描述的内容往往涉及多个对象间的复杂关系，式3.1中的约束条件难以被直接转化为函数形式的数学表达式。因此，该问题很难通过凸优化或反向传播传统的优化方法求解。

3.2.2 领域知识转化为约束

LASIN 方法的主要思路是使用*反绎逻辑推理*（logical abduction）将一阶逻辑表达的领域知识用于约束机器学习的假设空间。直观来看，LASIN 的思想可与近端梯度下降（proximal gradient descent）[5] 方法进行类比，它在机器学习的迭代优化过程中能够不断地使用反绎逻辑推理将优化结果尽可能地限制于领域知识所允许的空间中。

具体来说，LASIN 采用了第3.1节中介绍的主动的假设检验策略[136] 进行学习，它可以细分为 4 个步骤：1) 根据观测到的现象主动生成一系列假设；2) 根据这些假设，再次进行观测并收集必要的相关信息；3) 检验这些额外信息是否能够佐证第 1 步生成的假设；4) 根据检验结果选择接受、排除或更新自己的假设。

在 LASIN 的学习过程中，这 4 个步骤分别对应于：1) 通过反绎逻辑推理得到候选假设空间；2) 在该假设空间中使用机器学习方法得到一个假设模型；3) 在训练数据中对该假设模型进行检验，判断它是否能够对数据分布进行准确的

建模；4) 根据检验结果选择排除或接受该假设模型，若排除假设模型，则重新进行第 1 步并更新假设空间。整个过程如算法3.1所示。

反绎逻辑推理

LASIN 方法利用 KB 生成候选假设空间的主要方法是*反绎逻辑推理*（logical abduction）[127]，这一过程体现在算法3.1的第 10 行。它可以跟据 KB 和上一轮迭代中所学模型 h_t 在训练数据中的预测误差 *error* 产生下一轮机器学习目标假设空间 $\mathcal{H}_{t+1}$。

与从一般规律推出具体事实的*演绎推理*（deduction）和从具体事实推出一般规律的的*归纳推理*（induction）不同，Peirce 提出反绎推理的目标是从具体观测事实中归因出另一部分具体事实，以作为对观测现象的解释[127]。

在人工智能领域中，反绎逻辑推理最常见的形式化是 Kakas 等人[73] 提出的反绎逻辑程序设计（Abductive Logic Programming，ALP），它能够根据输入的反绎逻辑理论（Abductive Logic Theory）(KB, A, IC) 和逻辑事实 O 推理得到另一组具体事实 Δ，其严格形式化定义如下：

定义 3.1 (反绎逻辑程序设计) 给定一个*反绎逻辑理论* (KB, A, IC)，其中 KB 是一段由形如式1.1的逻辑规则所构成的逻辑程序，A 是 KB 中定义的可*反绎谓词*（abducible predicates）或可反绎事实（abducible facts）集合，IC 是一组由逻辑规则构成的*完整性约束*（integrity constraints）。当给定一组*观测事实*（observed facts）O 时，反绎逻辑推理的目标是输出一个关于 A 的具体逻辑事实集合 Δ，被称为*反绎事实*（abduced facts），它必须满足：

1. KB 与 Δ 能够对 O 进行*解释*，即 $KB \cup \Delta \models O$；
2. Δ 满足 IC 所定义的约束，即 $KB \cup \Delta \models IC$；
3. Δ 与 KB 不产生冲突，即 $\mathrm{Con}(KB \cup \Delta)$。

直观来看，反绎逻辑推理的目标是当观察到现象 O 时，根据领域知识 KB 使用 Δ 进行合理地解释。例如，我们考虑如下逻辑规则构成的反绎逻辑理论：

$$wet_shoes \leftarrow wet_grass. \tag{3.3}$$

$$wet_grass \leftarrow rain_last_night. \tag{3.4}$$

$$wet_grass \leftarrow sprinkler_was_on. \tag{3.5}$$

$$false \leftarrow rain_last_night \wedge sprinkler_was_on. \tag{3.6}$$

规则3.3-3.5为反绎逻辑理论中的 KB，它们分别定义了事实“鞋子被打湿”（*wet_shoes*）与“草地湿润”（*wet_grass*）的原因，其中前者可观测事实 O；规则3.6则为完整性约束 IC，它表示“昨晚下雨”（*rain_last_night*）和“洒水器打开”（*sprinkler_was_on*）不可能同时成立；而其中的可反绎事实 A 则为造成观测事实出现的两种可能原因 {*rain_last_night*, *sprinkler_was_on*}。

当观测到 *wet_shoes* 为真时，ALP 会通过反绎推理以对其归因。通过规则3.3，ALP 会发现只有当 *wet_grass* 为真时，观测现象才可能成立。接下来，它会继续对 *wet_grass* 成立的原因进行查询，并通过规则3.4-3.5反绎推理出 *rain_last_night* 和 *sprinkler_was_on* 均有可能成立。而若我们拥有另一个观测事实——“昨晚未下雨”，则根据完整性约束规则3.6，必有 *sprinkler_was_on* 为真。因此，该反绎推理过程最终得到的 $\Delta = \{sprinkler_was_on\}$。

基于反绎推理的假设空间约束

对于 LASIN 来说，反绎逻辑推理所使用的 KB 就是学习任务中引入的领域知识；可反绎事实 A 则为当前假设空间中所有可能的假设模型 h；而观测事实则是上一轮学得模型 h_t 在训练数据中造成的误差 $error_t$；完整性约束 $IC = \Phi$。当 LASIN 进行至第 t 轮迭代时，反绎逻辑推理所求的解释 Δ 则为根据以上条件所反绎推理得到的新假设空间 $\mathcal{H}_t$。

因此，LASIN 的反绎逻辑推理过程 abduce($error_t, KB$) 可以被形式化为如下形式：

$$\begin{aligned} & KB \cup \mathcal{H}_{t+1} \models O. \\ \text{s.t.} \quad & \mathrm{Con}(KB \cup \mathcal{H}_{t+1}). \end{aligned} \tag{3.7}$$

上面提到，LASIN 中用于反绎推理的观测事实 O 为上一轮模型 h_t 在 D 中造成的误差，我们将其记为 $error_t = \{\varepsilon_1, \ldots, \varepsilon_m\}$，其中

$$\varepsilon_i = \mathrm{difference}(h_t(\boldsymbol{x}_i), y_i), \tag{3.8}$$

它表示 h_t 在样本 $\langle \boldsymbol{x}_i, y \rangle$ 中造成的误差。那么，根据 KB 和 ε_i 反绎推理出来的一个候选假设模型 $\hat{h}_t^i$ 可表示为：

$$KB \cup \hat{h}_{t+1}^i \models \varepsilon_m. \tag{3.9}$$

因此，式中所求的 $\mathcal{H}_t$ 即为

$$\mathcal{H}_{t+1} = \{\hat{h}^1_{t+1}, \ldots, \hat{h}^m_{t+1}\}, \tag{3.10}$$

此外，我们还可以利用这些 $\{\hat{h}^i_{t+1}\}$ 作为基向量张成一个更大的假设空间：

$$\mathcal{H}_{t+1} = \mathrm{span}(\hat{h}^1_{t+1}, \ldots, \hat{h}^m_{t+1}), \tag{3.11}$$

其中 $\mathrm{span}(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n)$ 表示由 $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$ 张成的空间。

当得到新的假设空间 $\mathcal{H}_{t+1}$ 以后，LASIN 便可以使用常规的机器学习方法学得假设模型 h_{t+1}。若令 $\mathcal{H}_{t+1} = \{\hat{h}^1_{t+1}, \ldots, \hat{h}^m_{t+1}\}$，那么机器学习的目标即是从 $\mathcal{H}_{t+1}$ 中筛选出经验风险最小的 h_{t+1}。由于 $\{\hat{h}^1_{t+1}, \ldots, \hat{h}^m_{t+1}\}$ 均通过式3.9中的反绎推理得出，根据定义3.1，$\forall h \in \mathcal{H}_{t+1}$ 均满足式3.1中的约束。

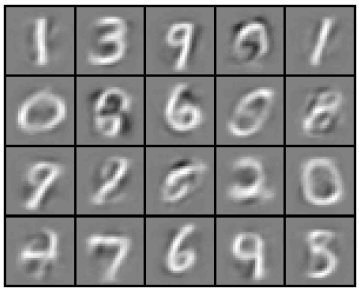
在许多情况下，对 $\mathcal{H}_{t+1} = \{\hat{h}^1_{t+1}, \ldots, \hat{h}^m_{t+1}\}$ 进行筛选并不一定是最佳选择，因此更好的选择是通过式3.11张成一个更大的空间。不过，$\mathrm{span}(\hat{h}^1_{t+1}, \ldots, \hat{h}^m_{t+1})$ 可能结构依然十分复杂。为了令 $\mathcal{H}_t$ 中能够进行效率较高的机器学习，我们还可以在使用 $\{\hat{h}^1_{t+1}, \ldots, \hat{h}^m_{t+1}\}$ 张成空间的同时牺牲一定 $\mathcal{H}_{t+1}$ 关于 KB 的一致性，以换取假设空间更好的性质，例如凸性等等。

3.3 实验验证

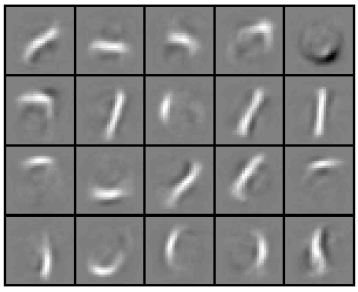
著名的人工智能学者侯世达曾提出：“人类的字符识别问题中包含着绝大部分人工智能的基本挑战”[63]，这其中就包括了在学习过程中对领域知识的运用。在学习认字时，人类能够自然而然地将“笔划”这一领域知识作为辅助约束运用在字符图片的分解、重组和识别过程。这种关于“笔划”的领域知识难以通过数学表达式描述，但却可以使用一阶逻辑语言进行描述。

本节的实验中，我们将 LASIN 应用于手写字符图片的特征提取。实验中，我们还通过引入不同的领域知识检验它们在该问题中的效果。下面的几小节中将分别对该问题中 LASIN 方法的具体实现、实验设置与实验结果进行介绍。

(a) MNIST 和 Omniglot 手写字符数据集



(b) 稀疏编码输出的结果



(c) LASIN 输出的结果

图 3.1 手写字符图片中不采用和采用领域知识约束时的学习结果

3.3.1 LASIN 在图像特征提取中的应用

问题定义

该实验的目标是借助领域知识对手写字符图片进行非监督的特征学习（feature learning），并检验引入/未引入领域知识时学得特征在分类任务上的效果。图3.1 展示了一个从手写字符图片中以领域知识作为约束进行概念抽取的例子。其中图3.1(c)中的结果是引入关于笔划的领域知识后，从 MNIST[95] 手写数字图片中提取的特征；而图3.1(b)则是未引入领域知识时，直接使用稀疏编码（Sparse Coding）[96] 学得的特征。

本小节中将这种使用一阶逻辑表达的领域知识作为辅助约束的特征提取问题称为概念提取（concept extraction）问题，它的严格形式化定义如下：

定义 3.2 (概念提取) 该任务的输入为一系列训练样本 $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$ 以及一个领域知识库 KB，其中 $\boldsymbol{x}_i \in \mathbb{R}^n$，$KB$ 则由一组一阶逻辑公式构成。该任务的输出为由一组特征构成的字典 $\boldsymbol{b} = \{b_1, b_2, \ldots, b_s\}$，其中的 $b_j \in \mathbb{R}^n$，它们必须能够精确地重构出 D 中的样本。此外，每个 b_j 必须与 KB 中所描述的概念一致，且 $\exists p \in KB$ 使得 $KB \models p(b_j)$，这里的 p 为 KB 中的谓词。

可以看到，概念提取问题中 KB 对学习的约束比式3.1中定义的一致性更强，它要求存学得的特征满足 KB 中某个谓词 p 的定义。

概念提取问题可以看作是对表示学习[10] 的一种扩展。表示学习是一类特征

学习方法，其目标是从数据中自动提取出有用的信息作为特征表示，以提高机器学习模型的分类精度和运算速度。在手写字符识别等实际问题中，表示学习已得到广泛应用，例如上面例子中提到的稀疏编码[96,125]、流形学习[174]和深度神经网络等等[19]。然而，由于表示学习一般被形式为统计优化问题，因此很难引入一阶逻辑表示的领域知识[46]。

该任务中，关于“笔划”领域知识可以用自然语言进行描述：每个字符均由一个或多个笔划构成，每个笔划表现在图片中是一段连续不中断的墨迹，且笔划中可能出现转折。该知识同样可以通过一阶逻辑语言描述为 KB_{stroke}：

$$character(C) \leftarrow C = \{S_1, S_2, \ldots\} \wedge stroke(S_1) \wedge \ldots. \tag{3.12}$$

$$\begin{aligned} stroke(S) \leftarrow & S = \{P_1, P_2, \ldots\} \\ & \wedge sub_strk(P_1, P_2, P_3) \wedge sub_strk(P_2, P_3, P_4) \wedge \cdots. \end{aligned} \tag{3.13}$$

$$\begin{aligned} sub_strk(A, B, C) \leftarrow & ink(\boldsymbol{AB}) \wedge ink(\boldsymbol{BC}) \\ & \wedge angle(\boldsymbol{AB}, \boldsymbol{BC}) < \alpha). \end{aligned} \tag{3.14}$$

其中原子公式 $stroke(S)$ 表示序列 $S = \{P_1, P_2, \ldots\}$ 构成一条笔划轨迹，其中每个点 $P_i = (X_i, Y_i)$ 为图片中的一个坐标；原子公式 $sub_strk(A, B, C)$ 表示 A、B 和 C 三个点构成一个子笔划，其中向量 $\boldsymbol{AB}$ 和 $\boldsymbol{BC}$ 分别是子笔划中的前后两段；原子公式 $ink(\boldsymbol{AB})$ 的赋值为 *true* 当且仅当图片中以 A、B 为端点的向量 $\boldsymbol{AB}$ 是一段连续的墨迹，没有出现中断的空白；$angle(\boldsymbol{AB}, \boldsymbol{BC})$ 是一个计算 $\boldsymbol{AB}$ 与 $\boldsymbol{BC}$ 向量夹角大小的函数；$\alpha > 0$ 是一个用于限制 $\boldsymbol{AB}$ 与 $\boldsymbol{BC}$ 夹角大小的参数，本章实验中均设置为 $\alpha = \pi/2$。

领域知识转化为约束

上面所定义的 KB_{stroke} 即可看作一个反绎逻辑理论：规则 3.12-3.14构成了领域知识；$\{stroke/1\}$ 即为可反绎谓词；完整性约束 $IC = \Phi$。对于手写字符图像中的概念提取问题来说，观测事实 O 即是“每个训练样例均为手写字符”，用 KB_{stroke} 中的谓词表示即是 $\forall \boldsymbol{x}_i \in D(character(\boldsymbol{x}_i))$。

对上面的反绎逻辑理论 (KB_{stroke}, A, Φ) 输入观测事实 $character(\boldsymbol{x}_i)$ 时，反绎逻辑程序可以一步步地验证 KB_{stroke} 中的定义，并尝试输出一个关于反绎谓词 $stroke/1$ 的具体事实集合 $\hat{h}_i = \{s_{i1}, s_{i2}, \ldots\}$，其中的每个 s_{ij} 均满足 $stroke(s_{ij})$。它的具体过程如下：

1. 当输入观测事实 $character(\boldsymbol{x}_i)$ 时，令 $j=1$，根据式3.12从图像 $\boldsymbol{x}_i$ 中采样一个满足 $stroke/1$ 定义的 s_{ij}；
2. 采样 s_{ij} 时，令 $k=1$，根据式3.13从 $\boldsymbol{x}_i$ 中采样三个满足 $sub_strk/3$ 的点 (p_k, p_{k+1}, p_{k+2})；
3. 令 $k=k+1$，直至图像中无法采样出满足式3.13和 3.14式中定义的下一个 p_k；
4. 令 $s_{ij}=\{p_0,\ldots,p_k\}$ 且 $j=j+1$，继续重复第 1 步直至图像中无法采样出满足式3.13中 $stroke/1$ 定义的下一个 $stroke(s_{ij})$；
5. 输出 $\hat{h}_i=\{s_{i1}, s_{i2}, \ldots\}$。

该反绎推理过程中会不断地使用 $ink/1$ 对输入图像中的墨迹进行采样，在实验中我们通过第4章中将要介绍的 KRL 方法实现。为了增加 p_k 序列采样的效率，我们通过递归实现的贪婪搜索定位序列中的下一个墨点 p_{k+1}。因此该采样算法在每个样本 $\boldsymbol{x}_i$ 上的时间复杂度为 $O(K\log K)$，其中的 K 为最多采样墨点个数。

直观来看，$\hat{h}_i$ 是反绎推理根据领域知识将 $\boldsymbol{x}_i$ 拆解而得到的“可能笔划”，而并非字符在书写过程中所使用的真正笔划。这是因为反绎推理只能根据领域知识对观测事实做出许多可能解释，而并非唯一的最佳解释[101,159]。因此，LASIN 根据 $\boldsymbol{KB}_{stroke}$ 找到的不能被称为“笔划”（strokes），至多只能称为可能的“书写原语”（written primitives）。此外，这也意味着若想真正找到最优的书写原语，我们还需借助其他机器学习方法。

基于稀疏编码的特征提取

在对每个样本 $\boldsymbol{x}_i$ 进行反绎逻辑推理后，LASIN 可以得到一个概念假设集合 $\mathcal{H}=\{\hat{h}_i\}$，它们分别是从每个样本中提取出来且满足 $\boldsymbol{KB}_{stroke}$ 的候选特征。

本章采用稀疏编码[96] 进行特征学习。稀疏编码是一种效率较高的特征学习方法，它的目标是从样本空间中学习一个字典作为特征，以使得每个原样本在经过该字典编码后所得到的特征向量均为稀疏向量。那么，LASIN 在手写字符特征提取任务中机器学习目标函数可以表示为：

$$\begin{aligned} &\min_{\boldsymbol{b},\boldsymbol{a}} \sum_k \|h^{(k)} - \sum_j a_j^{(k)} b_j\|_2^2 + \beta\|a^{(k)}\|_1 \\ \text{s.t.}\quad &\|b_j\|_2 \le 1, \forall j \in 1,\ldots,s. \\ &\forall h^{(k)}((h^{(k)} \in \mathcal{H}) \wedge (\boldsymbol{KB} \models stroke(h^{(k)}))). \end{aligned} \tag{3.15}$$

其中每个 $h^{(k)} \in \mathcal{H}$ 意味着 $h^{(k)}$ 是根据领域知识 KB 通过反绎推理从 x 中提取得到的假设概念；$stroke$ 是 KB 中定义“笔划”的谓词，($KB \models stroke(h^{(k)})$ 表示 $h^{(k)}$ 必须满足 KB 中关于 $stroke/1$ 的定义；$\boldsymbol{b} = \{b_1, b_2, \ldots, b_s\}$ 则是稀疏编码最终输出的**字典**（dictionary），其中的每个 $b_j \in \mathbb{R}^n$ 均为一个提取出来的特征，它们可看作经过编码后新特征空间中的基向量；$\boldsymbol{a} = \{a^{(1)}, a^{(2)}, \ldots, a^{(n)}\}$ 是字典 $\boldsymbol{b}$ 对 $h^{(k)}$ 的**编码**，即 $\boldsymbol{b}$ 所张成特征空间中每个 $h^{(k)}$ 的特征向量，它们的每一维 $a_j^{(k)}$ 的大小代表特征 b_j 在用于重构 $h^{(k)}$ 时的激活程度。

经过机器学习得到的 $\boldsymbol{b}$ 可以看作从假设空间 $\mathcal{H}$ 中总结出的最佳特征，也就是领域知识 KB_{stroke} 从数据集 D 学到的“最佳书写原语”。

然而，稀疏编码的优化过程会对特征进行重构，而并非直接从满足 KB 约束的 $\{\hat{h}_1, \ldots, \hat{h}_m\}$ 中进行筛选，因此它输出的 b_j 不能保证依然满足 KB 中对 $stroke/1$ 的定义。本章中选取它而不是其他特征选择（feature selection）方法作为 LASIN 的机器学方法主要出于以下两个原因：1、稀疏编码是一种无监督表示学习方法，而大部分特征选择方法需要使用监督信息。为了减少监督信息对特征提取造成的偏差，并使得提取得到的特征能够被迁移至不同的领域，无监督的稀疏编码是一种更好的选择。2、本节所进行的实验均采用像素表示的图像作为数据来源，其中每个样例特征向量中的一个维度代表图片中一个像素位置的灰度值。使用稀疏编码时，很容易进行“有/无反绎逻辑推理 + 稀疏编码”的对照实验设计。而特征选择算法很难适用于此类像素表示的数据，因而难以作为实验中的对照方法。

手写字符学习中的假设检验

当稀疏编码学得一组特征 $\boldsymbol{b}$ 后，LASIN 会检验其是否能够精确地重构原始数据，并根据检验结果选择输出 $\boldsymbol{b}$ 或对其进行更新。在手写字符图像特征学习任务中，该假设检验过程的步骤如下：

1. 利用 $\boldsymbol{b}$ 对 D 中的每个样本进行稀疏编码，该过程可以被描述为如下形式：

$$\arg\min_{\boldsymbol{a}} \sum_i \|\boldsymbol{x}_i - \sum_j a_j^{(i)} b_j\|_2^2 + \beta \|a^{(i)}\|_1$$

其中的 b_j 均来自于 LASIN 在上一轮学得的字典 $\boldsymbol{b}$；

2. 使用编码 $\boldsymbol{a} = \{a^{(i)}\}$ 对所有训练样本 D 进行重构，得到 $D' = \{\boldsymbol{x}'_i\}$，即：

$$\boldsymbol{x}'_i = \sum_j a^{(i)} \cdot b_j, \quad \forall \boldsymbol{x}'_i \in D$$

3. 计算每个样本的重构误差 $\varepsilon_i = \|\boldsymbol{x}'_i - \boldsymbol{x}_i\|_2$，当 ε_i 大于阈值 *threshold* 时将其归入重构失败样例集 *error*；
4. 若重构失败集合 *error* 不为空，则将其作为下一轮 LASIN 所使用的训练数据集 D，并在下一轮的稀疏编码时以当前字典为初始值重新学习；否则输出当前字典。

而在 LASIN 进行的前几轮，由于 $\boldsymbol{b}$ 为空或者质量较低，绝大部分样例均会被纳入重构错误集。对于样例数量较大的数据集，将它们全部用于反绎逻辑推理会消耗大量时间。因此，在 LASIN 的实现中我们对 *error* 集合先进行随机采样再进行随后的学习步骤。

3.3.2 实验设置

实验数据

本节实验中共使用了两个常见的手写字符数据集，每个数据集中样例的例子请分别见图3.1(a)和图3.3(a)，它们分别为：

- **MNIST**[95]：该数据集中有有 70000 张从 0 到 9 共 10 类手写数字图片，每张图片的大小为 28×28，其中 60000 张为训练样例，其余为测试样例。
- **Omniglot**[92]：该数据集包含了各种语言共 1628 类字符，其中每个类别仅 20 个样例，每张图片的大小为 105×105。为了令该数据集中提取的特征能被用于 MNIST 数据，我们将它的图片缩放为 28×28 大小。该数据集被分为 Omniglot_small_1 与 Omniglot_small_2 两部分，我们分别使用它们进行了实验，下文将其简写为 OS1 与 OS2。与 Lake 等人[92] 在原文中的方法不同，本小节的实验中没有使用笔迹数据，而仅用了图片。

对比方法

为了检验引入领域知识是否能带来帮助，本节的实验中对 4 种特征学习方法进行了对比：

- **LASIN-stroke**：该方法是本章提出的特征提取算法，下文图表中将其简写

为 stroke。它通过引入关于笔划的领域知识对稀疏编码进行约束。实验中，算法3.1的参数均通过交叉验证设定。为了检验引入领域知识后，LASIN 是否有能力从小数据中学习，在 MNIST 任务中，该方法仅对每个类别采样 100 个样例，共 1000 个样例进行训练。

- **LASIN-cluster**：它一个包括 LASIN-kmeans 和 LASIN-spectral 两种方法，下文图表中将它们分别简写为 kmeans 和 spectral，它们属于是 LASIN-stroke 的变体。与 LASIN-stroke 不同的是，它们在反绎推理时不引入前者关于笔划的领域知识，而是使用聚类将字符的形状分割为不同的部分，以用于对稀疏编码进行约束。这种领域知识同样可以通过一阶逻辑公式进行表达：

$$parts(C,S) \leftarrow ink_points(C,P) \wedge cluster(P,S,k). \tag{3.16}$$

其中 P 是字符 C 中的所有墨点（即图片中取值非 0 的像素位置），$S = \{s_1,\ldots,s_k\}$ 是对 P 进行聚类得到的簇，$cluster = kmeans$ 或 $cluster = spectral$ 分别代表两种聚类方法。kmeans 是一种基于欧式距离的聚类方法，而 spectral 使用的谱聚类则是一种基于流形的聚类方法。由于手写字符可以看作 1 维流形在 2 维平面中的嵌入，因此采用谱聚类假设十分自然。聚类算法无法像 LASIN-stroke 一样自动确定分割的“笔划”数目，在本小节的实验中我们均设置 $k = 2$。在 MNIST 任务中，这两类方法均使用了全部 60000 个训练样例。

- **稀疏编码（Sparse Coding）**：该方法没有引入任何领域知识，被作为基准方法与其他方法进行对比。在 MNIST 实验中它同样使用了全部 60000 个训练样本。下文图表中将其简写为 SC。

其中所有 LASIN 算法判断重构失败的阈值参数 *threshold* 均通过 10-折交叉验证选取。

我们分别以字典大小 $|\boldsymbol{b}| \in \{20,50,100,200\}$ 进行了实验。与数据原本的维度 784 相比，这些字典大小均偏小，这是因为构造字符的笔划种类其实并不多，因此它的编码长度不应该过长。

为了评估每种方法学得字典的优劣，我们采用 Raina[138] 等人提出的自学习（self-taught learning）方法对它们进行评估：每个学得的字典会被用于对所有样例进行编码，然后我们在编码后的训练数据中学习一个分类器，并在测试数据集中验证该分类器的精度。若分类精度越高，则说明该字典的质量越好，即从数据中提取出了更有意义的特征。

表 3.1 对比方法在不同任务中的自学习平均分类精度（%）

	$\|b\| = 20$				$\|b\| = 50$			
数据集	*SC*	*stroke*	*kmeans*	*spectral*	*SC*	*stroke*	*kmeans*	*spectral*
MNIST	90.37	**91.66**	90.48	90.82	93.89	**94.88**	94.47	94.79
OS1	**16.46**	16.24	15.83	15.33	21.27	**21.82**	20.92	21.09
OS2	15.12	16.65	17.37	**17.43**	22.61	**22.90**	21.12	21.05
	$\|b\| = 100$				$\|b\| = 200$			
MNIST	95.23	**96.27**	96.19	96.05	95.77	**97.01**	96.91	96.97
OS1	23.48	**24.74**	23.26	23.45	25.40	**26.64**	25.94	26.37
OS2	24.74	**25.07**	24.10	23.98	25.63	**26.29**	26.00	26.21

基于自学习方法，我们一共设计了两类任务对所有对比方法进行检验：

1. **手写字符图片分类**：该任务对每个数据集进行自学习，然后测试学得字典在分类任务中的表现，它可以用于衡量学得字典在原领域中的好坏。
2. **字典迁移**：该任务中，源领域（source domain）中学得的字典将被用于对目标领域（target domain）中的样例进行编码，然后通过自学习的分类精度在目标领域上检测源领域字典的质量。本实验中采取 MNIST 和 Omniglot_small_1 互相作为源领域与目标领域进行迁移。

以上两个实验中自学习时采用的分类器均为 libSVM[17] 实现的线性核支撑向量机，其参数均由交叉验证确定。本实验中没有采用更复杂的分类模型，因为我们希望在实验中尽可能控制分类器本身能力对分类精度的影响，以确保模型效果的提升由是引入不同领域知识而带来。实验中的稀疏编码以及 LASIN-spectral 和 LASIN-kmeans 中的聚类方法均由 mlpack 工具包[26] 实现。反绎逻辑程序则通过 SWI-Prolog 工具包[171] 和 Prolog 语言实现。

3.3.3 实验结果

手写字符图片分类

自学习分类任务的结果请见表3.1，其中显著优于其他对比方法的结果由黑体标出。可以看到，在 MNIST 任务中，基于 LASIN 的三种方法均好于未引入领域知识的稀疏编码。而在 Omniglot 字符的两个任务中，所有方法的预测精度均较低。这是因为它们的类别太多，且训练数据量十分稀少。总的来看，引入

表 3.2 基于分块的稀疏编码自学习平均分类精度（%）

$\|\boldsymbol{b}\|$	5×4	13×4	20×4	25×4	50×4	100×4
准确率	85.02	91.62	92.69	93.94	94.98	96.41

“笔划”领域知识的 LASIN-stroke 在所有任务中取得了最好的效果。该结果验证了引入领域知识后 LASIN 的有效性。

不过，因为基于 LASIN 的概念提取方法均是依据领域知识将字符分割后才进行稀疏编码，每次进行统计学习时它们所使用的训练样例会比未引入领域知识的稀疏编码更多。而在表示学习时也可以通过对图像分块以增广训练样例，我们采用该方法在 MNIST 数据中也进行了对比实验。该实验中，MNIST 中的图片被均等地分割为四块，其中每个分块大小为 14×14。然后我们在分块的数据集中使用稀疏编码，学得了字典大小为 $|\boldsymbol{b}'|\in\{5,13,20,25,50,100\}$ 的字典。利用它们对原 28×28 的数据进行编码时，将会分别对图像的四个分块进行编码，再将其拼合为一个特征向量，最终被用于学习分类器。因此这些分块后学得字典对数据的总编码长度为 $|\boldsymbol{b}|\in\{20,52,80,100,200,400\}$。通过对比表3.2与表3.1中的结果中可以看出，在总编码长度相同的情况下，LASIN 获得的结果仍然超过了稀疏编码。即便以分块的编码长度进行比较，LASIN 在只使用 1/4 编码长度的情况下仍然可以获得与之相近的表现。此外，LASIN 在提取特征时只使用了 1000 个训练样例，其数据量远远少于稀疏编码。这些结果都验证了引入领域知识的确能够提高学得字典（特征）的质量。

表3.1中的结果还展示了不同领域知识对学得字典的影响。在多数情况下 LASIN-spectral 的结果均优于 LASIN-kmeans，这是可能是因为谱聚类假设对于字符图像更加合理。但是在 Omniglot 的两个任务中，尽管绝大多数情况下 LASIN-stroke 的表现比稀疏编码更好，但是 LASIN-kmeans 和 LASIN-spectral 却有时比稀疏编码的效果更差，这是因为我们将聚类个数 k 设置为 2。对于 MNIST 里的阿拉伯数字手写字符，这样的假设还算合理；但是在 Omniglot 中绝大多数字符的笔划数目都超过了 2。与之相对的，LASIN-stroke 所引入的领域知识最为合理，因此在所有任务中均取得了不错的表现。因此，选取合适的领域知识能够促进机器学习的效果，而选择不合适的领域知识则可能对机器学习的效果造成损害。

表 3.3 将字典进行迁移后的自学习分类精度（%）

	$\|\boldsymbol{b}\| = 20$				$\|\boldsymbol{b}\| = 50$			
任务	*SC*	*stroke*	*kmeans*	*spectral*	*SC*	*stroke*	*kmeans*	*spectral*
M2O	19.36	**20.69**	19.09	18.60	20.39	**23.38**	20.86	21.09
O2M	87.72	**88.14**	87.98	88.06	93.08	**93.69**	93.44	93.67
	$\|\boldsymbol{b}\| = 100$				$\|\boldsymbol{b}\| = 200$			
M2O	20.43	**23.19**	20.93	21.48	18.62	**24.63**	22.57	22.82
O2M	96.03	96.00	**96.11**	95.81	96.11	**96.39**	96.19	96.22

字典迁移

迁移学习任务的结果被展示在表3.3中。其中 M2O 表示以 MNIST 为源领域，以 Omniglot_small_1 为目标领域；O2M 则反之。从结果中可以看到，在此任务中表现最好的依然为引入“笔划”领域知识的 LASIN-stroke。通过对比表3.3 和表3.1，可以发现在编码长度较短的情况下，M2O 进行字典迁移的效果甚至比直接在 Omniglot 中进行自学习的效果更好。这可能是因为 MNIST 拥有更丰富的数据，在编码长度较少的情况下能够提取出得更有意义的特征和概念。该现象也表明，在某些情况下将已有的模型进行重用，可能比从头开始学习一个全新的模型更好[177]。

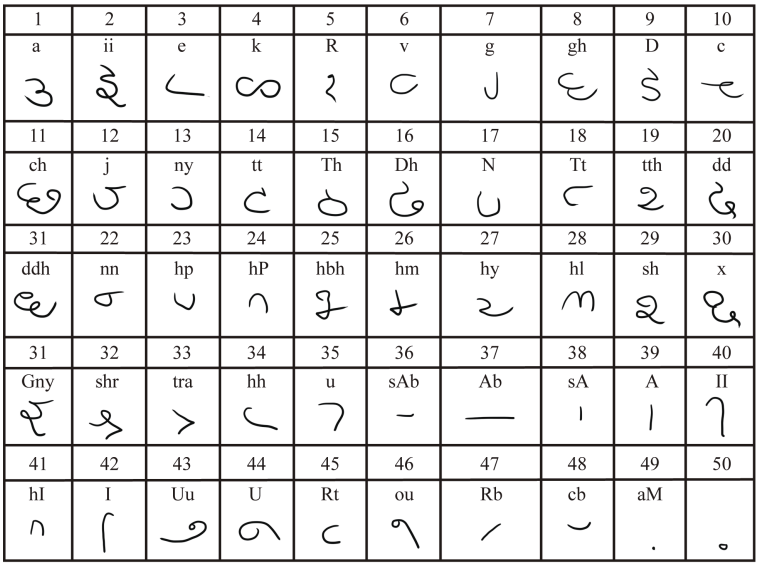
3.3.4 讨论

除了将 LASIN 应用于机器学习任务，我们还设计了另一个实验以检验它是否能够通过引入领域知识辅助约束以学得更符合人类认知的特征。

该实验所使用的数据集为 **HPL-Devanagari**[11]。此数据集收集了超过 100 个印度语使用者所书写的 111 种印度 Devanagari 语手写字符，其中每种字符的样例平均有 270 个。每个 Devanagari 字符均由一系列笔划构成，图3.2[85] 展示了人类在书写该语言时使用的所有笔划。该实验中，我们设置提取特征个数（字典大小）为 $|\boldsymbol{b}| = 200$，并使用式3.12-3.14中定义的规则作为 LASIN 的领域知识。未引入领域知识的稀疏编码与 LASIN 中使用的稀疏编码均采用了相同的参数。

图3.3(c)和图3.3(d)分别展示了 LASIN 和稀疏编码输出的字典结果；图3.3(b)则展示了 LASIN 在一次反绎逻辑推理后输出的假设概念 $\{\hat{h}_1, \ldots, \hat{h}_m\}$。通过对比 LASIN 和稀疏编码的输出结果，明显可以观察出前者的结果更接近

Devanagari中的笔划集合-1



Devanagari中的笔划集合-2

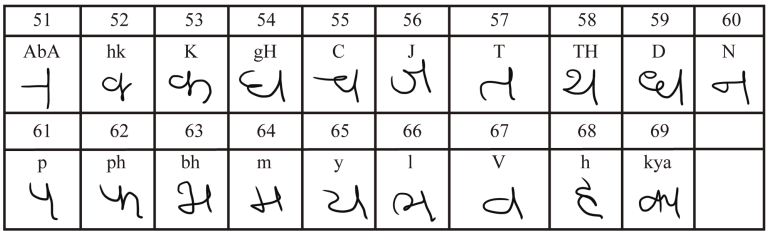
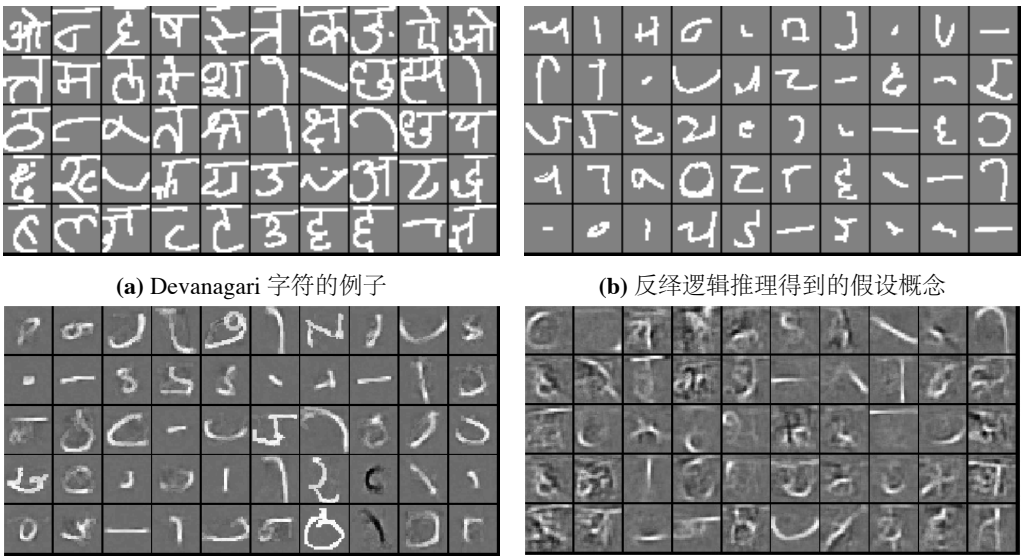


图 3.2 人类定义的两类 Devanagari 笔划[85]

于图3.2。也就是说，引入“笔划”领域知识的 LASIN 的确能够提取出与人类认知更相似的特征。这是因为它能够通过反绎逻辑推理，将假设空间限制在一个更符合 KB 定义的局部，所以稀疏编码所学得的假设模型不至于偏离 KB 的约束太远。将图3.3(b)和图3.3(c) 进行对比可以发现，稀疏编码后学得的特征比反绎推理得到的假设概念更加模糊，这是因为它对后者进行了优化，使之比 $\{\hat{h}_1,\ldots,\hat{h}_m\}$ 更具代表性，即能够通过稀疏编码准确地重构更多字型。

3.4 小结

与2章中所面对的问题不同，本章关注的是当领域知识无法一次性增广为质量较好的训练样本时，如何将领域知识以约束的形式嵌入机器学习过程中。本章提出的 LASIN 方法是一种利用一阶逻辑领域知识辅助约束机器学习方法，



(a) Devanagari 字符的例子 **(b)** 反绎逻辑推理得到的假设概念

(c) LASIN 提取的特征 **(d)** 稀疏编码学得的特征

图 3.3 印度语手写字符书写原语学习结果

它可以通过反绎逻辑推理将一阶逻辑表示的领域知识转化为机器学习中对假设空间的约束。在 LASIN 的机器学习优化过程中，它可以不断收集信息以评估每轮学得模型的效果，并根据这些信息对领域知识生成的约束进行调整。本章将 LASIN 应用于手写字符图像特征学习任务中，实验结果显示，使用领域知识辅助机器学习的 LASIN 能够取得比其他方法更好的效果。

本章的主要工作：

- W.-Z. Dai and Z.-H. Zhou. Combining logic abduction and statistical induction: Discovering written primitives with human knowledge. **In: Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'17)**, San Francisco, CA, 2017, pp.4392-4398. （*CCF-A* 类会议）

第四章　一种机器学习驱动的领域知识精化方法

4.1　引言

前两章中的工作主要关注如何利用一阶逻辑表示的领域知识辅助机器学习，而本章则考虑如何将机器学习技术应用于一阶逻辑知识精化任务。

知识精化是人工智能的核心任务之一，它的目标是从数据中学习一组描述目标概念的逻辑规则。它获得的模型拥有良好的可理解性，因此往往被用于科学发现等需要人机交互的学习任务中[78,79]。

当训练数据仅为一阶逻辑公式和具体事实构成的集合时，知识精化问题可以通过基于逻辑表示的机器学习方法求解，例如归纳逻辑程序设计（ILP）[107]等等；而当这种训练数据中存在噪声时，则可以通过概率归纳逻辑程序设计（PILP）[34]或统计关系学习（SRL）[54]求解。

然而，绝大部分实际任务中均存在非逻辑符号表示的数据，将它们一一标注为逻辑符号往往需要消耗大量人力，因此上面所提到的方法均难以适用。但是，在这样的领域中同样存在着许多需要知识精化的学习任务。例如从天文图片中学习行星运行规律、从股票数据中学习投资策略等等。

本章提出的 KRL（Konwledge Refinement by Learning）是一种机器学习驱动的知识精化方法。该方法将知识精化问题转化为一个机器学习任务，实现了一般数据中的知识精化。KRL 首先使用反绎逻辑推理，调用机器学习模型从数据中自动抽取与目标概念相关的逻辑事实，并以这些事实对领域知识进行增广。随后，KRL 会使用基于一阶逻辑表示的机器学习方法对增广的领域进行知识精化，最终输出可理解的一阶逻辑规则作为模型。

本章的实验中将 KRL 应用于计算机视觉问题，提出了 Logical Vision 方法。人工图像与真实图像中的实验均验证了 Logical Vision 只需消耗少量样本，即可以从图片中学习出由一阶逻辑规则表达的形式化定义。并且，它学得的知识能够作为新的领域知识被直接重用于更复杂的视觉学习或其它推理任务。

4.2 KRL 方法

KRL 方法的基本思想是使用机器学习模型将样本转化为增广的领域知识，再使用基于一阶逻辑表示的机器学习方法对增广的领域知识中进行精化。本节将先对 KRL 问题进行定义，再介绍 KRL 方法的具体步骤，最后介绍一中基于 KRL 的视觉知识精化方法，它可以从图像中实现一阶逻辑领域知识的精化。

4.2.1 问题定义

KRL 的目标是从训练数据和领域知识中使用机器学习进行知识精化。根据第2章2.2.1小节的定义，知识精化任务可以表示为:

$$B \cup H \models E. \tag{4.1}$$

其中知识精化任务的输入 B 为背景知识，E 为关于目标概念的训练样本；任务的输出 H 是假设模型。对于常规的知识精化任务来说，B、E 和 H 均为形如式1.1的一阶逻辑规则所构成的逻辑程序。

而对于 KRL 来说，输入数据中还可能包含非逻辑符号表示的样本。因此，本章定义 KRL 所输入的背景知识为 $B = KB \cup D$，其中 KB 是由一阶逻辑公式构成的领域知识库，$D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$ 为训练样本，其中每个 $\boldsymbol{x}_i \in \mathcal{X}$，而 $\mathcal{X}$ 为输入空间。$E = \{y_1, y_2, \ldots, y_m\}$ 则是对应于每个样本的标记，其中 $y_i \in \mathcal{Y}$，而 $\mathcal{Y}$ 为标记空间。由于 D 中可能不含逻辑符号，且 KB 不足以进行关于目标概念的知识精化，因此 KRL 需使用机器学习模型 $f : \mathcal{X} \mapsto \Gamma$ 实现从数据 D 中提取增广领域知识的基本逻辑事实，其中 Γ 表示 KB 中所有可能具体事实的集合，即 KB 的 Herbrand 域（Herbrand universe）[43]。KRL 的任务目标仍是进行知识精化，所以它的输出 H 依然是由一阶逻辑规则构成的集合。基于以上符号，KRL 的学习目标可以表示为:

$$KB \cup D \cup \{f\} \cup H \models E. \tag{4.2}$$

4.2.2 机器学习驱动的知识精化

KRL 的大概过程如算法4.1所示。它首先根据领域知识 KB 借助 f 从训练样本 D 中抽取与目标概念相关的逻辑事实集合 F，然后用这些逻辑事实增广领域

算法 4.1 KRL

1: **算法输入：**背景知识 $B = KB \cup D \cup \{f\}$；训练样例 E
2: **算法过程：**
3: **for** $\boldsymbol{x}_i \in D$ **do**
4: // 根据式*4.3*进行反绎推理，从原始数据中提取逻辑事实
5: $F_i = krl_abduce(\boldsymbol{x}_i, KB, f)$
6: **end for**
7: // 利用提取出的逻辑事实对 KB 进行增广
8: $F = \cup_{i=1}^{m} F_i$
9: $B' = KB \cup F$
10: // 根据式*4.1*使用基于逻辑表示的机器学习方法对 B' 进行知识精化
11: $H = learn(B', E)$
12: **算法输出：** H

知识 KB，最终使用基于逻辑表示的机器学习方法对增广后的领域知识进行精化。下面的小节中将对 KRL 进行详细介绍。

基于机器学习的领域知识增广

为了使用训练样本 D 对领域知识 KB 进行增广，KRL 的第一步是使用机器学习模型从样本中提取必要的逻辑事实（见算法4.1中第 5 行），而这一过程可通过反绎逻辑推理完成。

如第3章中所介绍，反绎推理（abductive reasoning）的目的是对观测现象做出解释，而这些被推导出解释均为具体事实（grounded facts）。在 KRL 中，反绎推理的目标是根据领域知识 KB，推导出与领域知识一致的具体事实 F，以对每个样本 $\boldsymbol{x}_i \in D$ 中所展示的现象做出解释。该过程的目标可以形式化为：

$$KB \cup F \models D. \tag{4.3}$$

该反绎推理可以使用定义3.1所介绍的反绎逻辑程序设计（ALP）[73] 实现，它使用的领域知识为 KB，已观测现象为 D，完整性约束 $IC = \Phi$，反绎推理所得的结果 $\Delta = F$。

然而，4.3中的已观测现象 D 仍然为 $\mathcal{X}$ 空间中的样本，不一定拥有逻辑表示。因此，我们需要借助式4.2中的机器学习模型 f 对其进行采样并转化为逻辑事实。具体来说，对每个样本 $\boldsymbol{x}_i \in D$，式 4.3中的反绎逻辑推理可以借助机器学习模型 f 转化为如下形式：

$$KB \cup F \models f(\boldsymbol{x}_i) \tag{4.4}$$

算法 4.2 krl_abduce

算法输入：机器学习模型 f，领域知识 KB，训练样本 $\boldsymbol{x}_i$
算法过程：
// 生成初始假设
$t = 0$
$\Delta_t = \Phi$
// 使用 f 对 $\boldsymbol{x}_i$ 进行随机采样观测
$O_t = f(\boldsymbol{x}_i)$
while $t < T$ **do**
  // 使用机器学习模型 f 对 $\boldsymbol{x}_i$ 进行更多采样观测
  $O_{t+1} = O_t \cup f(\boldsymbol{x}_i)$
  $t = t + 1$
  // 使用检验 Δ_t 是否与观测结果 O 一致
  if $KB \cup \Delta_t \models O_t$ **then**
    Break // 跳出 *while* 循环
  end if
  // 根据式*4.4*使用反绎逻辑推理更新假设 Δ_t
  $\Delta_t = abduce(KB, O_t)$
end while
算法输出： $F_i = \Delta_t$

其中 $f(\boldsymbol{x}_i)$ 为机器学习模型 f 对 $\boldsymbol{x}_i$ 进行采样所得的具体逻辑事实构成的集合，它事实上可看作一种通过机器学习模型对 $\boldsymbol{x}_i$ 进行的观测。观测的目的不一定是将 $\boldsymbol{x}_i$ 预测为某个特定的类别，它也可以是使用 f 对 $\boldsymbol{x}_i$ 的局部进行采样而得到的关于某些基本概念（primitive concepts）逻辑事实，而这些基本概念则应当能够被用于构建增广领域知识的 F。

与 LASIN 相似，KRL 中使用的反绎推理也采用了第3.2节所介绍的主动假设检验策略[136]。它的第一步是使用领域知识 KB，主动对 $\boldsymbol{x}_i \in D$ 生成逻辑事实表示的符号假设 Δ，第二步为使用机器学习模型 f 对 $\boldsymbol{x}_i$ 进行采样观测，其观测结果表示为 $O = f(\boldsymbol{x}_i)$。第三步则由 ALP 根据观测结果 O 对符号假设 Δ 进行验证。若 $KB \cup \Delta \models O$ 则表示该假设与 KB 一致，那么便输出 $F_i = \Delta$；否则根据验证过程中所发现的错误对 Δ 进行修改，并再次通过 f 对 $\boldsymbol{x}_i$ 进行更多的采样观测并予以检验。该过程重复进行，直至从 $\boldsymbol{x}_i$ 中成功反绎推理出与 KB 一致的 F_i，如算法4.2所示。

KRL 的反绎推理过程是先由 KB 主动生成假设，为了验证假设是否成立，KRL 才对 $\boldsymbol{x}_i$ 使用机器学习模型 f 进行采样。该过程可与人类的视觉过程进行类比：想象一个人站在一副巨大的壁画面前，由于画面过于庞大，他无法一次性看清所有的部分，每次只能观察到壁画的一个局部。为了辨别画作是否属于某幅自己在美术书上见过的名画，他必须通过不断地移动自己的视线进行观

```
prove([], Prog, Prog) ← .
prove([Atom|As], Prog1, Prog2) ←
    metarule(Name, MetaSub, (Atom ← Body), Order)
    ∧ Order
    ∧ save_subst(metasub(Name, MetaSub), Prog1, Prog3)
    ∧ prove(Body, Prog3, Prog4)
    ∧ prove(As, Prog4, Prog2).
```

图 4.1 元解释学习（MIL）使用的元解释器

察。而最有效率的方式则是在观察时，根据已有的知识猜测整体画面的形式，然后针对自己的猜测主动选择下一个采样观察的局部，不断地排除或修正自己的猜测，最终形成一个足够正确（即多次检验后依然无法排除）的假设，接受它并完成辨别。

因此，通过反绎逻辑推理，KRL 能够实现效率较高的领域知识增广。简单来说，KRL 使用领域知识主导机器学习模型对样本的采样，然后以采样的结果为基础排除、修改或接受由领域知识生成的符号假设。它实现了按需进行的特征提取，减少了大量冗余操作。

基于机器学习的领域知识精化

一旦将领域知识增广为 $B' = KB \cup F$，4.2式中的 KRL 目标便可转化为：

$$B' \cup H \models E. \tag{4.5}$$

它与式4.1着相同的形式。此时，KRL 便能够使用任意基于逻辑表达的机器学习方法实现知识精化，例如 ILP[107] 和 PILP[34] 方法等等。

本章中主要使用元解释学习（Meta-Interpretive Learning，MIL）方法[114]进行知识精化。该方法是 ILP 领域最新的进展之一，它是一种由元解释器（meta-interpreter）[114] 实现的反绎逻辑推理方法。一般的一阶逻辑元解释器可以通过后向链[14] 推理以试图证明由一阶逻辑原子公式表示的已观测事实。在推理过程中，元解释器会不断地在领域知识中查找与当前证明目标形式相同的一阶逻辑规则，然后通过反绎推理获取证明该公式时需要的前提，并将这些前提作为对已观测事实的解释而输出。

元解释学习通过类似的技术对满足目标概念的样例进行证明，但它在证

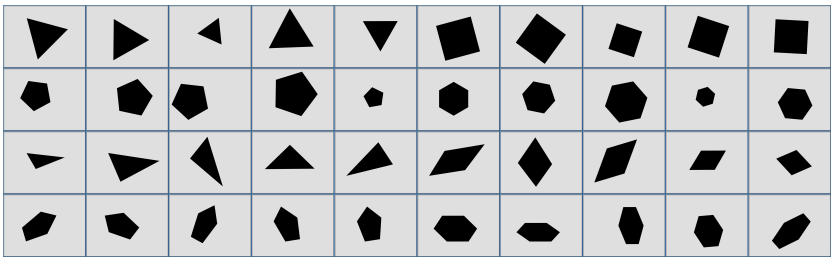


图 4.2 正多边形与非正多边形

明过程中所查找的“解释”是二阶逻辑[43]规则，它们也被称为元规则（meta-rule）。最终反绎推理得到的解释则为二阶逻辑规则的具体形式——即一阶逻辑规则。图4.1展示了元解释学习所使用的二阶反绎逻辑推理元解释器。

简而言之，元解释学习将式4.1中的知识精化任务转化为一个二阶逻辑中的反绎推理任务，它将满足目标概念的训练样例 E 作为已观测事实，将待学习的假设模型 H 作为对 E 的解释，然后通过反绎逻辑推理找到符合式4.1的 H，最终达到一阶逻辑规则学习——即知识精化的目的。

4.2.3 基于 KRL 的视觉概念精化

本章中，我们将 KRL 应用于计算机视觉领域并提出了 Logical Vision 方法，它的目标是从图像中对领域知识进行精化。为了对 Logical Vision 进行更好地介绍，本小节将以一个简单的几何问题为例：图4.2展示了两类多边形的图片，其中前两行为正多边形，后两行为非正多边形。任务目标是从这些样例中学习出“正多边形”（regular polygons）的概念。

根据式4.2中对 KRL 的定义，该例子中的领域知识 KB 需要包含一些必要的基本几何概念，例如关于点、线、面和夹角的定义等等；D 为图4.2中的图片；训练样例 E 是关于这些图片是否为正多边形的标记，例如“$y_3 = true$”和“$y_{28} = false$”等等。而需要学得的假设模型 H 则是能够用于定义“正多边形”这一几何概念的一阶逻辑公式。

领域知识增广

为了从图像中学习出关于正多边形的概念，我们必须引入一些最基本的几何概念作为背景知识，例如点和线段等等。当提取出图片中的点和线段以后，便可以通过它们将图片中的多边形表示为逻辑符号。

学习正多边形概念的 Logical Vision 的领域知识 KB 中包含了三个基本概念 $\{polygon/2, edge/1, edge_point/1\}$，这三个谓词分别用于表示图片中的多边形、边界以及边界上的点。关于多边形的逻辑事实可表示为 $polygon(Pol_i, [Edge_1, \ldots, Edge_n])$，其中 Pol_i 为多边形的编号，它在增广 KB 时的反绎推理的过程中被自动生成；$[Edge_1, \ldots, Edge_n]$ 是一个构成多边形的线段序列。每条线段 $Edge_j$ 是该多边形的一条边，它由一个二元组 (P_1^j, P_2^j) 构成，其中每个 $P_k^j = (X_k^j, Y_k^j)$ 分别代表这条线段的起点和终点的坐标。

关于多边形的领域知识可通过如下一阶逻辑规则表示：

$$\begin{aligned} polygon(Pol_i, [Edge_1, \ldots, Edge_n]) \leftarrow & \ edge(Edge_1) \wedge \ldots \wedge edge(Edge_n) \\ & \wedge connected(Edge_1, Edge_2) \wedge \ldots \wedge connected(Edge_n, Edge_1). \end{aligned} \tag{4.6}$$

其中的 $edge/1$ 是反绎逻辑程序中的可反绎谓词，它可用于判断两点 (P_1^j, P_2^j) 之间是否存在一条线段作为分隔目标对象（多边形）与背景的边界；而 $connected/2$ 是一个用于检验两条线段是否首尾相连的谓词。

通过该规则，我们可以形式化地定义多边形的概念。但真正的挑战在于如何使用该领域知识从图像中发现多边形以对 KB 进行增广。根据第4.2.2小节所介绍的反绎推理过程，我们首先需要使用领域知识生成关于 $polygon$ 的假设，此外还需引入机器学习模型 f 对生成的假设在图像中进行检验。

根据背景知识式4.6中的定义，若想将图片转化为关于 $polygon$ 的事实并对“正多边形”概念进行学习，必须从图片中提取出 $edge$ 的逻辑事实。而 $edge$ 所代表的“边”是由连续“点”构成的直线线段，关于它的领域知识和检验过程可简单地被描述为如下两条一阶逻辑规则：

$$edge([P_1, P_2], 0) \leftarrow midpoint(P_1, P_2, P) \wedge edge_point(P). \tag{4.7}$$

$$\begin{aligned} edge([P_1, P_2], N) \leftarrow & midpoint(P_1, P_2, P) \wedge edge_point(P_1) \wedge edge_point(P_2) \\ & \wedge N' = N - 1 \wedge edge(P_1, P, N') \wedge edge(P, P_2, N'). \end{aligned} \tag{4.8}$$

这两条规则定义了一个关于“边”概念的二元谓词 $edge/2$，它的第一个参数 $[P_1, P_2]$ 表示 P_1 与 P_2 之间存在一条多边形的边界；它的第二个参数 N 表示验证该事实所需要的进行的采样观测次数；$midpoint(P_1, P_2, P)$ 表示 P_1 和 P_2 的中点为 P；而 $edge_point(P)$ 则是算法4.2中的机器学习模型 f，它能够采样 P 邻域中的像素，并判断 P 是否处于多边形的边界上。

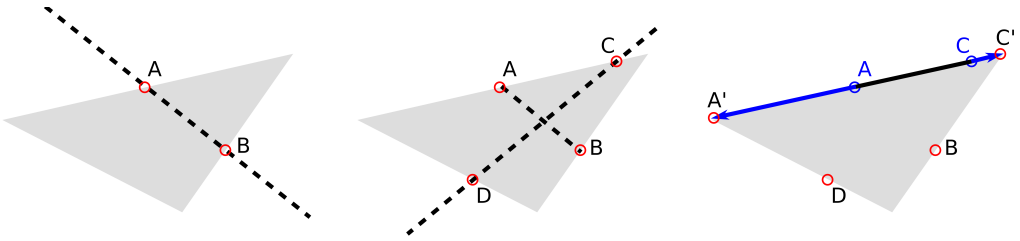


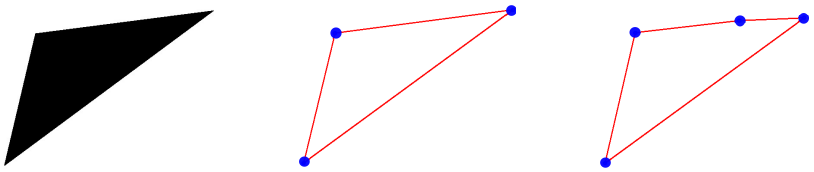
图 4.3 基于多边形概念的领域知识增广示意图

事实上，规则4.7-4.8构成的领域知识递归地定义了一个第4.2.2小节中描述的主动假设检验过程。其中规则4.8表示，若 P_1P_2 是一条边界，那么首先必须采样观测 P_1 和 P_2，并验证它们均为多边形与背景之间的分界点；然后，对于它们的中点 P，还必须递归地采样观测 P_1P 和 PP_2 是否均为边界。规则中的 N 则控制递归采样的次数，规则4.7表示当 $N = 0$ 时不需要进行递归，只需调用机器学习学习模型对中点 P 进行采样观测即可。

图4.3中的例子展示了这一过程：图4.3(a)表示机器学习模型随机采样观测所得的 O_0 为 $\{edge_point(A), edge_point(B)\}$，因此猜测了一个关于“边”的解释 $\Delta_0 = edge([A, B])$；图4.3(b)表示猜测 $edge([A, B])$ 未通过检验，因此机器学习模型重新从图像中采样观测了 C、D 两个边界点；图4.3(c)中更新了关于边的假设得到 $edge([A, C])$，它最终通过了检验，并被输出作为领域知识的增广。

在上文描述的领域知识增广过程中，为了将基于符号表示的反绎逻辑推理应用在基于像素表示的图像中，我们使用机器学习模型 $edge_point/1$ 对图片中的像素点进行分类。当 $edge_point(P)$ 输出大于 0 时表示 P 的确是目标对象与图片背景间的分界点，而输出小于 0 时则表示该点不属于分界点。它是整个 Logical Vision 在进行领域知识增广过程中，唯一直接与图片进行交互的函数，即 KRL 反绎推理过程中的采样观测函数 f。对于简单的问题，例如图4.2中不存在噪声的人造图片，我们可以通过计算机视觉中的 Sobel 算子实现 $edge_point$ 函数，该算子可以计算图像 A 中的梯度 G：

$$G(A) = \sqrt{G_x(A)^2 + G_y(A)^2} \tag{4.9}$$



(a) 原始图片 **(b)** 未出现噪声 **(c)** 出现噪声

图 4.4 基于多边形概念的领域知识增广的噪声

其中：

$$G_x(A) = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A \tag{4.10}$$

$$G_y(A) = \begin{bmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * A. \tag{4.11}$$

上式中的 $*$ 表示计算机视觉中常用的二维卷积操作，G_x 和 G_y 分别表示图像 A 在 x 轴和 y 轴方向上的梯度大小。

除了以上这些领域知识以外，我们还引入了谓词 $rmv_rdndnt(Pol_1, \gamma, Pol_2)$ 去除多边形中可能多余的边。这是因为基于采样的领域知识增广在将图像解释为线段构成的多边形时，有可能会提取出多余的边，图4.4展示了这一现象。$rmv_rdndnt/3$ 可以删去 Pol_1 中夹角 α 满足 $\pi-\gamma<\alpha<\pi+\gamma$ 的顶点以获得 Pol_2，以达到消去噪声的目的。但需要注意的是，尽管我们将其作为领域知识引入知识精化任务中，但是否将 $rmv_rdndnt/3$ 纳入假设模型、以及 γ 的具体取值大小，均必须由 Logical Vision 根据数据中噪声的情况自主学习。

在显微镜图片或天文图像等真实数据中，图片里往往存在着大量噪声。由于设备、光照和拍摄角度等因素的不同，图片的色彩、对象的形态以及图像中的噪点分布均有可能出现大量变化。因此仅仅使用简单的 Sobel 算子检测对象与背景间的分隔点并不实际。为了完成真实图像中的领域知识增广，Logical Vision 能够采用更复杂的统计学习模型实现 $edge_point/1$ 谓词，例如预训练的图像分割模型等等[176]。

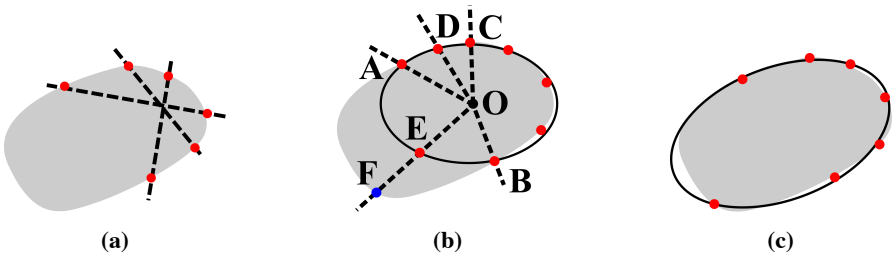


图 4.5 基于椭圆形概念的领域知识增广示意图

另外，真实图像中目标对象的形状往往更加复杂，它们经常由曲线构成。因此，仅使用多边形进行表示的效率可能并不高。与多边形相比，圆和椭圆能够通过更少的参数近似地描述许多几何形状。因此，Logical Vision 也引入了关于这两种图形的几何概念作为领域知识。借助它们，Logical Vision 可以将目标对象表示为逻辑事实 $circle(Centre, Radius)$ 与 $elps(Centre, Parameter)$，二者分别表示圆形和椭圆形。其中 $Centre = [X, Y]$ 是圆或椭圆的中心，$Radius$ 是圆的半径，$Parameter = [A, B, Tilt]$ 为椭圆形的参数，A 和 B 分别表示它的长轴与短轴长度，$Tilt$ 是椭圆长轴的倾斜角度。

圆与椭圆同样可以通过基于反绎推理的领域知识增广方法从图像中提取，图 4.5简要描述了这一过程：图4.5(a)中通过 *edge_point* 采样得到了 5 个目标对象的边界点；图4.5(b)中 Logical Vision 根据这五个点生成了一个关于椭圆形的假设，并在椭圆形的边界上进行采样以验证该假设。但是，在采样过程中发现原本应当是目标对象边界的点 E 并不属于边界点，因此主动从 OE 的延长线上采样得到了边界点 F；图4.5(c)中则根据所有在验证过程中采样得到的边界点生成了新的椭圆假设，更好地描述了目标对象的形状与位置。

除了简单的几何图形，Logical Vision 还可以从图像中提取其他概念作为增广 KB 的 F。得益于一阶逻辑的强大表达能力，Logical Vision 能够引入更多的领域知识，并自动采样观测并提取出更复杂的对象。以这些关于对象形状和位置作为基本事实，它推导出更多复杂的视觉概念。例如通过对象形状内部的像素点的颜色分布推导出高光的位置，借助更高级的机器学习模型对几何形状内的纹理进行分类等等。

算法 4.3 $Metagol_{NT}$

算法输入：增广的领域知识 $B' = KB \cup F$；元规则 M；训练样例 E；关于数据集噪声程度的参数 ν；迭代最大轮数 T
算法过程：
// 初始化最高得分 *max_score* 和最佳假设模型编号 *best_model*
$max_score = 0$
$best_model = 1$
while $t < T$ **do**
    // 根据参数 ν 随机从 E 中抽取一组训练样例
    $Tr_t = randSample(E, \nu)$
    // 将剩下的数据作为验证集
    $Ts_i = E - Tr_i$
    // 调用标准的元解释学习算法 *Metagol* 从 Tr_t 中学习概念假设 H_t
    $H_t = Metagol(B' \cup M, Tr_t)$
    // 在验证集 Ts_t 中对 H_t 进行检验
    $score_t = evaluate(H_t, Ts_t)$
    if $max_score < score_t$ **then**
        $best_model = t$
        $max_score = score_t$
    end if
end while
算法输出：H_{best_model}

有噪声情况下的知识精化

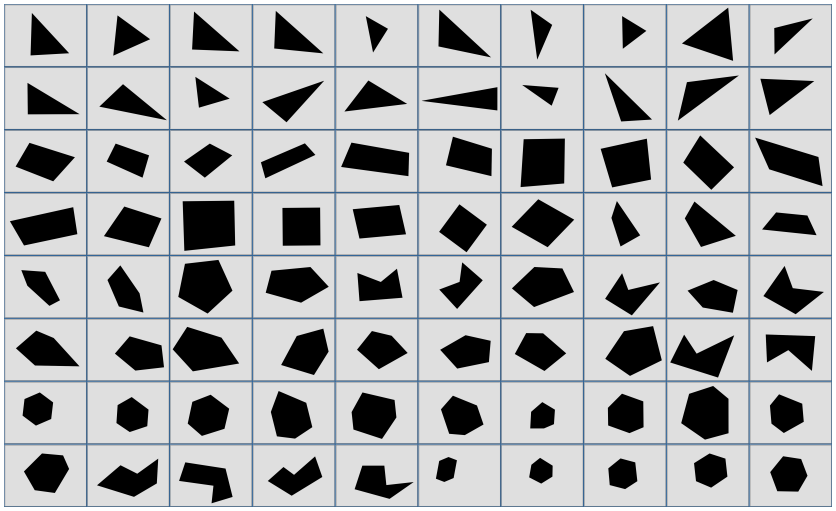
由于图像中通常存在噪声，从图片中提取所得的逻辑事实 F 中可能也包含噪声。为了控制图像噪声对学习造成的影响，本小节提出了一种通过对训练样本进行随机采样进行知识精化的元解释学习算法 $Metagol_{NT}$，它伪代码如算法4.3所示。

标准的元解释学习算法 *Metagol* 在实现中假设领域中不存在任何噪声，因此当增广的领域知识和元规则分别为 B' 和 M 时，它学得的假设 H 必须满足：

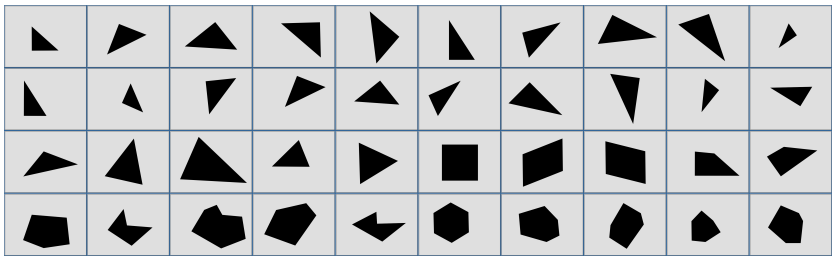
$$\forall e^+ \in E(B' \cup M \cup H \models e^+) \wedge \forall e^- \in E(B' \cup M \cup H \not\models e^-) \tag{4.12}$$

其中 M 为元解释学习所使用的二阶元规则。

由于基于反绎推理和采样的增广领域知识 F 可能存在噪声，所以很可能会出现 $\exists e^+ \in E(B' \cup M \cup H \not\models e^+)$。在这种情况下，*Metagol* 无法学习任何假设模型，最终只能输出 $H = \Phi$。因此，$Metagol_{NT}$ 选择从训练样例 E 中随机采样部分样例作为训练集进行 *Metagol* 学习，并使用剩余的训练样例验证其效果，最终输出性能最好的假设模型。其中对训练样例进行采样的比例基于用户预设的参数 ν，它反映了用户对领域中噪声水平的评估。



(a) 三角形、四边形、五边形及六边形



(b) 直角三角形、非直角三角形、四边形、五边形及六边形

图 4.6 人造多边形数据（正多边形数据请见图4.2）

4.3 实验验证

本小节分别在人工图像与真实图像中的视觉知识精化任务中对 Logical Vision 方法进行了检验，下面将对这两个实验进行分别介绍。

4.3.1 人造数据集中的实验

实验数据

在该实验中，我们使用 Inkscape[①]随机生成了 3 个有标记的多边形数据集，它们分别对应了 3 个多边形相关概念的知识精化任务。为简便起见，每张图片均为灰度图（png 格式），且每张图片中仅包含一个多边形。每个任务的数据集均被随机分为 5 等分，实验均采用 5 折交叉验证的方式进行评估[②]。

[①] http://inkscape.org

[②] 实验中所有的数据和源代码已公开于 https://github.com/haldai/Logic-Vision

实验设置

下面将对这 3 个任务的目标以及用到的领域知识进行简单介绍:

1. **多边形概念学习：**此任务中，我们随机生成了三角形、四边形、五边形和六边形共 4 种多边形的图片，其中每种多边形均包含 40 个样例，共计 160 个样例。

   - 任务目标：该任务中的目标概念为谓词 *triangle*/1、*quadrilateral*/1、*pentagon*/1 以及 *hexagon*/1，分别对应于数据集中的 4 种多边形。
   - 领域知识：除了第4.2.3小节中介绍的几本几何概念定义（规则4.6至规则4.8）和多边形噪声去除谓词 *rmv_rdndt*/3，该任务中还引入了标准 Prolog 谓词 *length*/2，它可以计算列表 $[A, B, \ldots]$ 的长度。需要注意的是，$rmv_rdndnt(Pol_1, \gamma, Pol_2)$ 中的参数 γ 需要由 Logical Vision 自主从数据中学得。

2. **正多边形概念学习：**该任务的部分数据请见图 4.2，它由第 1 个任务中 4 种多边形的正规以及非正规形状组成。其中每种多边形的正规和非正规形状个包含 10 个样例，因此一共有 80 个样例。

   - 任务目标：数据集中的样例被分为正规与非正规两大类，目标概念为 *regular*/1，它必须能够定义所有的正 n 边形。
   - 领域知识：为了学习正多边形概念，除了任务 1 中的所有知识外我们还引入了多边形夹角的定义和线段长度的定义，它们分别由谓词 *angle_list*/2 和 *edgelen_list*/2 表示，可用于计算多边形中所有夹角和边长的大小。此外，还引入了谓词 $std_dev_bounded(L, \delta)$，它用来测量列表 $L = [X, Y, \ldots]$ 中所有数字的标准差是否超过 δ，而这里的参数 δ 并非事先设定，而需要由 Logical Vision 从数据中学习。

3. **直角三角形概念学习：**我们随机生成了 40 个直角三角形作为该任务的正样例。该任务的负样例由非直角三角形、四边形、五边形以及六边形构成，其中每种形状个 10 个。为了检验 Logical Vision 是否可以学得直角三角形的概念，我们以 40% 的概率生成带直角的四边形、五边形和六边形。

   - 任务目标：该任务的目标概念为 *right_tri*/1，它需要对直角三角形的概念进行定义。

- 领域知识：任务 1 中学得的多边形概念作为领域知识被重用于该任务。此外，我们还引入了谓词 $has_angle(L,\theta,\delta)$ 用来表示夹角列表 $L = [X,Y,\ldots]$ 中存在一个大小为属于范围 $\theta \pm \delta$ 的角，这里的参数 θ 和 δ 和任务 2 中的 δ 相同，均需要从数据中学习得出。

实验结果

下面将展示基于 KRL 方法的 Logical Vision 在这 3 个任务中知识精化所得的结果。在任务 1 中，关于三角形的定义它学会了如下模型：

$$triangle_2(A,C,H) \leftarrow rmv_rdndnt(A,B,C) \wedge list_length(B,H). \tag{4.13}$$

$$triangle_1(A,A2,B2) \leftarrow polygon(A,B) \wedge triangle_2(B,A2,B2). \tag{4.14}$$

$$triangle(A) \leftarrow triangle_0(A,0.04,3). \tag{4.15}$$

$$triangle_1(A,G) \leftarrow polygon(A,B) \wedge list_length(B,G). \tag{4.16}$$

$$triangle(A) \leftarrow triangle_1(A,3). \tag{4.17}$$

其规则4.15和规则4.17对目标概念 $triangle/1$ 进行了定义，而其他规则中的 $triangle_1/3$、$triangle_0/3$ 和 $triangle_0/2$ 则是 *Metagol* 在学习目标概念过程中自动发明的谓词。可以看到，这 5 条规则将图像中的三角形分成了两种情况。最简单的情况由规则4.16和规则4.17 定义，后者表示当 A 满足 $triangle_0(A,3)$ 时它是一个三角形；而规则 4.16中发明的 $triangle_0/2$ 事实上表示多边形 A 的边数为 G。因此，这两条规则说明“边数为 3 的多边形是三角形”。而规则4.13- 4.15则定义了另一种情况，观察到它引入了去噪谓词 $rmv_rdndnt/3$ 并自动学得了其中的参数。通过简单解读可知，这些规则表示“当出现噪声的情况下，移除所有夹角在 $(\pm 0.04)\pi$ 范围的冗余顶点后，边数为 3 的多边形是三角形”，而后者正是平面几何中对三角形的形式化定义。

对于任务 1 的其他几个目标概念，Logical Vision 也学得了如下类似于 $triangle/1$ 的结果：

$$quadrilateral_1(A,G) \leftarrow polygon(A,B) \wedge list_length(B,G).$$

$$quadrilateral(A) \leftarrow quadrilateral_1(A,4).$$

$$pentagon_1(A,G) \leftarrow polygon(A,B) \wedge list_length(B,G).$$

$$pentagon(A) \leftarrow pentagon_1(A,5).$$

$$hexagon_2(A,C,H) \leftarrow rmv_rdndnt(A,B,C) \wedge list_length(B,H).$$
$$hexagon_1(A,A2,B2) \leftarrow polygon(A,B) \wedge hexagon_2(B,A2,B2).$$
$$hexagon(A) \leftarrow hexagon_1(A,0.004,6).$$
$$hexagon_1(A,G) \leftarrow polygon(A,B) \wedge list_length(B,G).$$
$$hexagon(A) \leftarrow hexagon_1(A,6).$$

通过解读可知，它们均准确地学会了这几种多边形的几何定义。

在任务 2 中，Logical Vision 学得了如下关于正多边形的目标概念：

$$regular_poly_2(A,G) \leftarrow angles_list(A,B) \wedge std_dev_bounded(B,G). \tag{4.18}$$
$$regular_poly_1(A,A2) \leftarrow polygon(A,B) \wedge regular_poly_2(B,A2). \tag{4.19}$$
$$regular_poly(A) \leftarrow regular_poly_1(A,0.02). \tag{4.20}$$

其中规则4.18和4.19分别发明了两个谓词。通过对规则内容的解读可知，这三条规则代表“若多边形所有内角的标准差不超过 0.02π，则它是正多边形”。这正是对多边形概念“所有内角相等”的一种定量描述。此外，值得注意的是我们在领域知识中作为陷阱加入了关于多边形所有边长的谓词 *edge_len_list*/2，但 Logical Vision 并没有选择它而学习出“边长相等的多边形是正多边形”。这可能是因为数据集中的确出现了边长相近的非正多边形，因此在交叉验证中排除了这样的模型。

在任务 3 中，Logical Vision 学会了如下关于直角三角形的概念：

$$right_tri_3(A,G,H) \leftarrow angles_list(A,B) \wedge has_angle(B,G,H). \tag{4.21}$$
$$right_tri_2(A,A2,B2) \leftarrow polygon(A,B) \wedge right_tri_3(B,A2,B2). \tag{4.22}$$
$$right_tri_1(A,A2,B2) \leftarrow right_tri_2(A,A2,B2) \wedge triangle(A). \tag{4.23}$$
$$right_tri(A) \leftarrow right_tri_1(A,0.5,0.015). \tag{4.24}$$

可以看到，规则4.23对任务 1 中学得的概念 *triangle*/1 成功进行了重用，这些规则共同定义了“若 A 是一个三角形，且 A 所有夹角中存在一个夹角大小属于 $(0.5 \pm 0.015)\pi$，则 A 是直角三角形”。

以上结果均表明，基于 KRL 方法的 Logical Vision 能够从图像数据中对领域知识进行精化，且学得的知识能够与这些图形的平面几何概念一致。

表 4.1 不同多边形图片分类任务中的精度（%）

方法	*triangle*	*quadrilateral*	*pentagon*
HOG	83.27	75.87	73.49
dense-SIFT	81.96	66.29	63.98
LBP	87.03	69.15	66.98
CNN	91.31	75.00	75.00
H+d	82.02	75.00	76.25
C+d	82.02	75.00	76.25
C+L	87.43	74.92	76.25
C+d+L	82.00	75.00	76.25
Logical Vison	∞〃.〃	∃∃.∃▽	∞〃.〃
	hexagon	*regular_poly*	*right_triangle*
HOG	75.08	63.02	74.01
dense-SIFT	70.98	71.02	77.06
LBP	72.97	65.03	75.29
CNN	84.38	59.39	85.22
H+d	76.25	64.14	80.47
C+d	76.25	68.93	80.47
C+L	76.25	60.87	78.20
C+d+L	76.25	64.30	80
Logical Vison	∃∃.∃▽	∞〃.〃	∞〃.〃

讨论

以上实验表明，基于 KRL 方法的 Logical Vision 能够在这 3 个图像数据集中学习出关于多边形的概念，并且这些概念是符合平面几何定义的。那么，如果抛开这些知识的形式，是否能够通过计算机视觉中的机器学习方法准确地学习出关于这些概念的模型呢？

我们在同样的数据中进行了分类实验，并与常见的图像识别方法进行了对比。由于数据量较小，我们无法直接使用深度神经网络进行学习，因此选择采用非线性核的支持向量机（由 libSVM[17] 实现）+ 主流计算机视觉特征的方法进行实验。其中，支持向量机的参数均通过 5 折交叉验证选取，实验所采用的特征有如下几种：

- **HOG**（Histogram of Oriented Gradients）[28]：该特征是对象检测问题中最常用的特征算子之一，它通过统计图像中局部的梯度的直方图作为特征。

- **Dense-SIFT**（dense Scale Invariant Feature Transform）[100]: 这是一类常用于图像匹配和识别的特征算子。提取该特征时，图像会先被分割为一个较密集的网格，其中每个局部使用 SIFT 方法提取特征并构成特征向量。
- **LBP**（Local Binary Pattern）[124]：这是一类用于描述图像局部纹理特征的算子，它具有旋转不变性和灰度不变性等显著的优点。
- **CNN**（Convolutional neural network）[95]：卷积神经网络是一类前馈神经网络模型，由于它能够利用输入数据的二维结构，因此被广泛使用于计算机视觉领域，并在深度学习中取得了成功的应用[91,151]。本节的实验采用预训练的 CNN 深度神经网络 `imagenet-vgg-verydeep-16`[154] 对每张图片提取一个 4096 维的特征。该网络是从 ImageNet ILSVRC-2012 数据集中训练而得，它提取的特征也在许多其他数据集的图像分类任务中均取得非常好的效果，拥有优秀的泛化性能[164]。
- **特征组合**：除了使用单个特征对图像进行分类，本节的实验中还考虑了不同的特征组合。

分类实验的结果如表4.1所示。可以看到，基于 KRL 的 Logical Vision 在测试集上可以得到近乎完美的分类结果。未使用平面几何领域知识的计算机视觉模型则在学习这些概念时遇到了困难，特别是在关于“正多边形”的预测上，它们的表现仅仅比随机猜测更好一点。根据知识精化任务中的结果，KRL 学得的形式化定义均为正确的，但在对测试集进行预测时仍然出现了错误。通过对结果的检查，我们发现 Logical Vision 出现的分类错误均来自于提取多边形事实时所产生的噪声（见图4.4）。

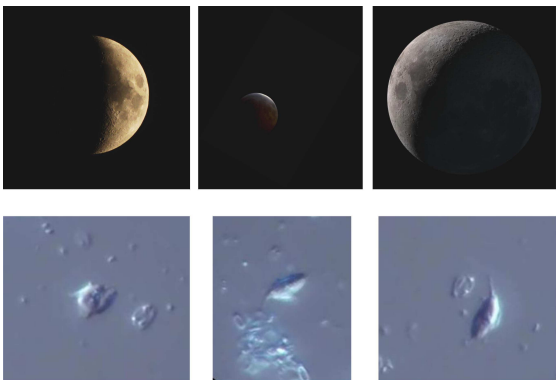
4.3.2 真实数据集中的实验

实验数据

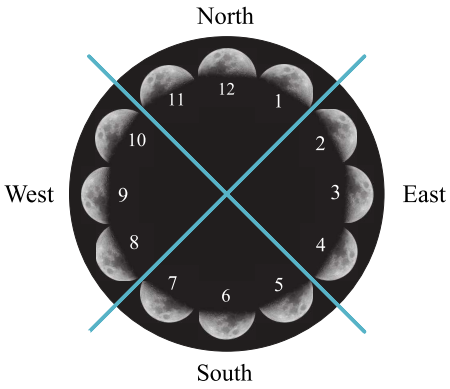
除了人工数据集中的实验，本章还对 Logical Vision 在真实图片中的知识精化能力进行了检验。如图4.7(a)所示，我们一共收集了两类图像以开展实验[①]:

- **Moons**：该数据集是从谷歌图像[②] 中收集的关于月球的天文照片。这些图像有着不同的来源，因此大小、格式不一，其中既有灰度图也有彩色图；
- **Protists**：该数据集为从一段原生生物（protist）显微镜视频中采样所得的图

[①]实验中所有的数据和源代码已公开于 https://github.com/haldai/LogicalVision2
[②]https://images.google.com/



(a) 月球与原生生物数据集



(b) 图像外光源方向检测

图 4.7 原生生物显微镜图像与月球天文照片

片，其中所有图片均为彩色图像。

如图4.7所示，该任务的目标是学习图像外光源的方向。为了对任务进行简化，我们将 12 个钟点方向平均分为 4 类，其中 $North = \{11, 12, 1\}$、$East = \{2, 3, 4\}$、$South = \{5, 6, 7\}$、$West = \{8, 9, 10\}$。其中每个钟点方向 30 张图片，因此每个数据集拥有 360 张图片。目标概念为 $light_source_angle(A, B, C)$，它表示图像中物体 A 受光源 B 照射，而光源的方向为 C。

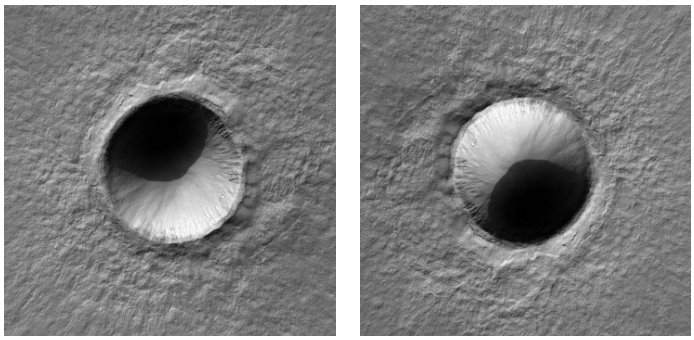
实验设置

在该任务中，Logical Vision 使用图4.5中的方法从图片中提取椭圆形以增广领域知识。当提取出图片中的对象后，它通过将对象 A 分割为两半以评估 A 中高光 B 的位置，该事实通过谓词 $high_light(A, B)$ 表示。同时，它还可以判断出高光 B 相对于物体 A 所在的角度 C，并用谓词 $clock_angle(A, B, C)$ 表示。在使用椭圆形概念增广领域知识时，Logical Vision 同样采用 $edge_point([X, Y])$ 找到分隔对象与背景之间的边界点，不过与多边形任务不同的是，此任务中它会从点 $[X, Y]$ 周围 10×10 邻域的像素提取颜色直方图特征，并调用在 5 张随机采样图片上预训练的机器学习模型判断 $[X, Y]$ 是前景与背景分界点的可能性，当概率大于 0.5 时返回 $true$，否则返回 $false$。

实验结果

从该任务中精化而得领域知识是一个反绎逻辑程序，从月球图片和显微镜图片中均成功得到了如下一阶逻辑规则集：

$$clock_angle(A, B, C) \leftarrow$$



(a) 环形山照片 **(b)** 翻转后的环形山照片

图 4.8 环形山照片数据集（来自于 NASA/JPL/University of Arizona）

$$clock_angle_1(A,B,D) \wedge light_source_angle(A,D,C).$$

$$clock_angle_1(A,B,C) \leftarrow$$

$$highlight(A,B) \wedge clock_angle_2(A) \wedge clock_angle_3(C).$$

$$clock_angle_2(Obj). \qquad clock_angle_3(Light).$$

规则集中的 *clock_angle_N* 均为知识精化过程中被发明出来的谓词，其中两个一元谓词 *clock_angle_2*/1 和 *clock_angle_3*/1 可以分别表示对象与光源的性质。当 Logical Vision 从图像中采样观测到物体上关于高光角度的事实 $clock_angle(A,B,C)$ 后，便可以通过以上规则反绎推理出光源的角度，该事实通过谓词 $light_source_angle(A,D,C)$ 表示。上面的规则表示“光源的角度与高光相对与物体的角度相同”。

此外，我们还尝试了从环形山图片中学习光源位置，该图像的例子如图4.8所示。与月球和原生生物不同的是，环形山的表面是下凹的，因此高光会出现在光源的对面。因此，我们加入了领域知识 $opposite_angle(A,B)$ 表示方向 A 与 B 是相对的。在该任务中学得的知识如下:

$$clock_angle(A,B,C) \leftarrow$$

$$clock_angle_1(A,B,D) \wedge clock_angle_4(A,D,C).$$

$$clock_angle_1(A,B,C) \leftarrow$$

$$highlight(A,B) \wedge clock_angle_2(A) \wedge clock_angle_3(C).$$

$$clock_angle_4(A,D,C) \leftarrow$$

$$light_source_angle(A,B,D) \wedge opposite_angle(D,C).$$

可反绎谓词	关于光反射的知识
prim(convex/1). prim(concave/1). prim(light_source/1). prim(light_angle/3).	highlight(X, Y):- contains(X, Y), brighter(Y, X), light_source(L), light_path(L, R), reflector(R), light_path(R, O), observer(O).
关于观测者的领域知识	
% "obj1" is an object discovered from image by LV; *% "obj2" is the brighter part of "obj1";* *% "observer" is the camera* contains(obj1, obj2). brighter(obj2, obj1). observer(camera). reflector(obj2). light_path(X, X). light_path(X, Y):-unobstructed(X, Z), light_path(Z, Y).	

图 4.9 用于反绎推理光源位置的逻辑程序

$$clock_angle_2(Obj). \qquad clock_angle_3(Light).$$

可以看到，在此任务中，Logical Vision 多学得了一个将对高光方向取反的谓词 *clock_angle*_4/3。这些规则表示“光源的角度与高光相对与物体的角度相反”。通过对比环形山和月球、显微镜图片中学得的规则可以发现，谓词 *clock_angle*_2/1 事实上代表对象反射表面的凹凸性（convexity）。

讨论

为了验证该任务中学得的知识能否重用，我们将它们应用于反绎逻辑推理，并使用它们判断未见图片中的光源位置。上面两个任务中分别学得的知识可以被等价地简化为如下形式：

$$\begin{aligned} &clock_angle(O, H, A) \leftarrow \\ &\quad highlight(O, H) \wedge convex(O) \wedge light_source(L) \\ &\quad \wedge light_source_angle(O, L, A). \\ &clock_angle(O, H, A) \leftarrow \\ &\quad highlight(O, H) \wedge concave(O) \wedge light_source(L) \\ &\quad \wedge light_source_angle(O, L, A1) \wedge opposite(A1, A). \end{aligned}$$

其中的 O 代表增广领域知识时从图像中提取的对象，H 代表高光，A 代表角度。这两条规则分别表示对象为凹凸表面时的高光与光源角度的关系。为了进

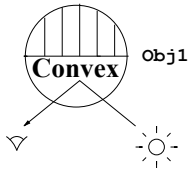
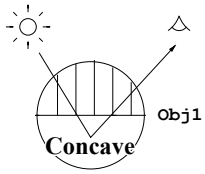
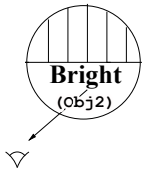
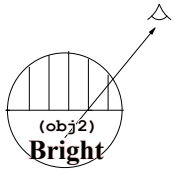
a)		light_source(light). light_angle(obj1, light, south). convex(obj1).
b)		light_source(light). light_angle(obj1, light, north). concave(obj1).
c)		light_source(obj2). light_angle(obj1, obj2, south). convex(obj1).
d)		light_source(obj2). light_angle(obj1, obj2, north). concave(obj1).

图 4.10 对图4.8(a)进行反绎逻辑推理得到的结果与示意图

行反绎逻辑推理，我们将这两条学得的规则加入了图4.9中的反绎逻辑程序。

在进行推理时，谓词 *convex*/1、*concave*/1、*light_source*/1 和 *light_angle*/3 被设置为可反绎谓词，它们分别表示物体表面是凸或凹、光源名字以及光源位置。当输入环形山图片时，反绎逻辑程序需要推理出这几个谓词的取值，以对图片中出现的现象进行解释。

图4.10展示了该反绎逻辑程序在输入类似图4.8(a) 样本时推理得到的结果以及示意图。其中图4.10(a) 中推理所得的事实表示光源位于图片南方，且物体表面是凸的；图4.10(b) 表示光源位于图片北方，且物体表面是凹的。

图4.10(c) 和图4.10(d) 则展示了更有趣的结果：它们均表示则表示环形山上的高光本身就是光源，因此光的传播方向既可以是向南，也可以是向北，结果完全取决于观察者所在的位置。

可以看到，KRL 学得的知识不仅能够用于对目标概念进行预测，还能从图

片中解读出歧义。这在目前其他形式的机器学习方法中还很难做到。

4.4 小结

本章中提出的 KRL 是一种机器学习驱动的领域知识精化方法。它将知识精化问题转化为一个机器学习问题。在学习过程中，KRL 能够使用机器学习模型对领域知识进行增广。随后，KRL 会使用基于一阶逻辑表示的机器学习从增广的领域知识中进行知识精化。为了验证 KRL 方法的有效性，本章还将它应用在计算机视觉领域，并提出了 Logical Vision 算法。实验显示，基于 KRL 的 Logical Vision 能够从图像中学得平面几何图形等视觉概念的形式化定义，且学得的知识能够被重用于图像分类或其他的复杂推理任务。

本章的主要工作：

- W.-Z. Dai, S. H. Muggleton, J. Wen, A. Tamaddoni-Nezhad and Z.-H. Zhou. Logical vision: One-Shot Meta-Interpretive Learning on Real Images. **In: Proceedings of the 27th International Conference on Inductive Logic Programming (ILP' 17)**, Orleans, France, 2017, pp.46-62. *(归纳逻辑程序设计领域旗舰会议)*
- W.-Z. Dai, S. H. Muggleton, and Z.-H. Zhou. Logical vision: Meta-interpretive learning for simple geometrical concepts. **In: Proceedings of the 25th International Conference on Inductive Logic Programming (ILP' 15)**. Kyoto, Japan, 2015, pp.1-16. *(归纳逻辑程序设计领域旗舰会议)*
- S. H. Muggleton, W.-Z. Dai, C. Sammut, A. Tamaddoni-Nezhad, J. Wen, and Z.-H. Zhou. Meta-interpretive learning from noisy images. **Machine Learning**. 107(7): 749-766, 2018. *(机器学习领域重要国际期刊)*

第五章　一种领域知识与机器学习互促结合框架

5.1　引言

前三章中分别研究了如何使用领域知识帮助机器学习以及如何使用机器学习来实现领域知识精化。本章希望探索的是如何实现领域知识与机器学习的互促结合，令它们在学习过程中相互帮助、共同提升。

正如第1章中介绍，人工智能领域中对机器学习与逻辑推理的研究绝大部分是分离的。但是许多学者们均认为，若此二者能够完美地相结合，则人工智能领域的发展将前进一大步[16,103,145,178]，因为人类智能正是这二者成功结合的一种体现。其中，人类的感知能力能够对图像、声音、触感等感知信息进行识别，这正对应于数据驱动的机器学习技术；而人类的逻辑推理能够对知识进行理解、推理和精化，它们则对应着逻辑程序和归纳逻辑程序设计等知识驱动的人工智能方法。

尽管将二者完美地相结合一直是人工智能领域中最核心的挑战之一，该问题的解决仍然面临着许多困难，本文的前三章中已或多或少的涉及。人工智能中知识驱动的推理和精化方法大多均基于一阶逻辑表示，而机器学习则大多被应用于从非逻辑符号表示的数据中进行对象识别。以往提出的结合方法则大多试图使用其中的一方来替代另一方，例如第1章中所介绍的概率逻辑程序（PLP）[29,120]和统计关系学习（SRL）[54]等等，它们要么重推理而轻学习，要么重学习而轻推理。

部分认知科学家指出，人类智能可通过反绎推理实现感知与推理的结合[101]。受该理论启发，本章提出的反绎学习（Abductive Learning）框架将机器学习与反绎逻辑推理相结合，以实现复杂学习问题中二者的相互促进。其中，机器学习可以对数据中的对象进行识别，将其转化为逻辑符号；反绎逻辑推理则能够判断识别出的符号是否与领域知识一致，并对识别错误的符号进行纠正。被纠正的符号则被转化为数据的标记，用于重新进行机器学习，以提升

机器学习模型的识别精度；而机器学习模型识别正确的符号也会被作为增广的领域知识，用于学习一阶逻辑规则，以实现知识精化。

基于反绎学习框架，我们将深度学习[55]与反绎逻辑程序设计（ALP）[73]相结合，提出了 NLM（neural logical machine）算法。人造数据集中的实验结果表明，NLM 能够同时进行领域知识的精化和机器学习。通过观察实验中间结果，我们发现反绎学习框架中的知识精化与机器学习的确可以相互促进。此外，学得的模型与精化而得的知识还可以在不同任务中得到重用。

5.2 反绎学习

反绎学习的目标是实现领域知识与机器学习的互促结合。其中领域知识的推理与精化均基于一阶逻辑表示；而机器学习则负责从数据中提取逻辑事实以对领域知识进行增广。本节将首先介绍反绎学习框架所面对的学习问题，再对其使用的技术进行介绍。

5.2.1 问题定义

反绎学习中机器学习模型的输出为一组用于增广领域知识的具体逻辑事实，它们被用于知识精化以对任务中的目标概念进行学习。并且，用于训练机器学习模型的标记——即这些被用于增广领域知识的逻辑事实——在数据中并未直接给出，它们需要由领域知识经过推理获得；知识精化所依赖的领域知识同样不完全，其中必须的一些逻辑事实则由机器学习模型从数据中提取。此任务可以被形式化为如下问题：

定义 5.1 (反绎学习) *反绎学习*的输入包含一组关于目标概念（target concept）C 的训练数据集 $D = \{\langle \boldsymbol{x}_1, y_1\rangle, \ldots, \langle \boldsymbol{x}_m, y_m\rangle\}$ 以及领域知识 KB。其中样本 $e_i = \langle \boldsymbol{x}_i, y_i\rangle$，且 $\boldsymbol{x}_i \in \mathcal{X}$ 是属于特征空间 $\mathcal{X}$ 的原始数据；$y_i \in \{true, false\}$ 是 $\boldsymbol{x}_i$ 关于目标概念 C 的标记；KB 则是由一阶逻辑公式构成的集合。其中目标概念 C 可以由领域中一组基本概念（primitive concepts）$\mathcal{P} = \{p_1, p_2, \ldots\}$ 描述，其中的 p_k 均为 KB 中定义的符号。*反绎学习*的目标是学习一个假设模型 $H = p \cup \Delta_C$，其中：

- *识别模型* $p : \mathcal{X} \mapsto \mathcal{P}$ 是一个从原始特征空间到基本概念的映射，它能够从样本 $\boldsymbol{x}_i$ 中提取一组关于基本概念的逻辑事实，以对领域知识进行增广；

- 知识模型 Δ_C 是一个由逻辑公式构成的集合，它能够与领域知识和被机器模型提取的概念一起对目标概念 C 进行定义；

它们满足：

$$\forall \langle \boldsymbol{x}, y \rangle \in D \quad (KB \cup \Delta_c \models p(\boldsymbol{x}) \wedge y) \tag{5.1}$$

可以看到，该问题主要存在以下两个难点：

1. 大部分机器学习模型的训练均需要直接的监督信息，例如对样本的标记。然而反绎学习中的机器学习问题中并不存在这种信息，它必须借助领域知识通过逻辑推理才能得到；
2. 知识精化和推理只能对基于逻辑表示的知识进行加工。但除了领域知识以外，另一部分重要信息却存在于原始数据中。若想以它们为原材料进行知识推理或精化，则必须学会如何从这些数据中识别出逻辑符号。

以上两个过程互相依赖，缺一不可，因此反绎学习比常规的机器学习问题更加困难。

5.2.2 反绎学习框架

如图5.1所示，反绎学习框架主要由机器学习（machine learning）、一致性优化（consistency optimization）以及反绎逻辑推理（logical abduction）组成。

图中的机器学习部件（machine learning）主要用于学习关于基本概念 $\mathcal{P}$ 的识别模型 p。由于它的输出必须是语义明确的逻辑符号，因此反绎学习中使用的一般为监督学习。值得注意的是，数据中并不存在关于 p 的直接监督信息，因此训练机器学习模型所使用的标记必须通过反绎逻辑推理来得到。

反绎逻辑推理部件（logical abduction）则既用于知识精化，也用于反绎推理出训练样本中可能存在的基本概念符号（primitive symbols），以作为训练识别模型的监督信息。在刚开始学习时，由于机器学习模型的训练并不充分，它识别出来的基本概念符号 $p(\boldsymbol{x}_i)$ 可能拥有错误，因此本文将它们命名为伪标记（pseudo-labels）。具体来说，反绎逻辑推理的工作过程如下：当输入训练样本 $\langle \boldsymbol{x}_i, y_i \rangle$ 时，它需要根据识别模型输出的伪标记 $p(\boldsymbol{x}_i)$、y_i 以及领域知识 KB 来推理并精化出 Δ_C；而当 $p(\boldsymbol{x}_i)$ 中有错误时，它需要通过反绎推理对 $p(\boldsymbol{x}_i)$ 进行纠正，并将纠正后的伪标记符号集用于重新训练识别模型 p。

由于样本中可能存在的伪标记符号集 $p(\boldsymbol{x}_i)$ 存在不确定性，且反绎逻辑推理的结果不唯一（请见第3章），这种学习模式十分困难。

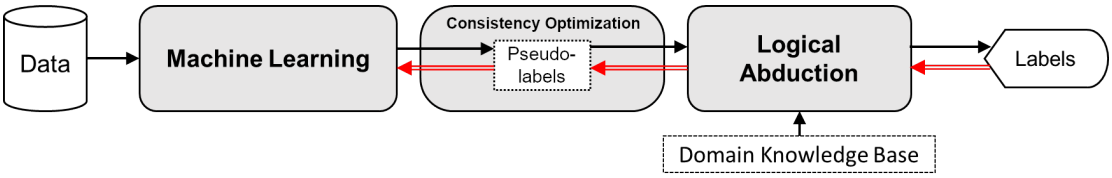


图 5.1 反绎学习框架

反绎学习试图通过一致性优化（consistency optimization）来解决以上问题。注意到样本标记 $Y = \{y_1, \ldots, y_m\}$ 和领域知识 KB 是学习过程中唯一可利用的监督信息，因此我们需要根据它们对所有样本中可能出现的伪标记符号集合 $p(X) = \cup_i p(\boldsymbol{x}_i)$ 和知识模型 Δ_C 进行同时优化，使得它们能够满足式5.1中的目标。换句话说，对于训练数据集 D，反绎学习需要根据领域知识 KB 对假设模型 $H = p \cup \Delta_c$ 进行优化，以最大化 H 与 D 一致的样本个数。该目标可以被形式化为如下表达式：

$$\max_{H=p\cup\Delta_C} \quad \mathrm{Con}(H \cup D) \tag{5.2}$$

其中 $\mathrm{Con}(H \cup D)$ 表示 D 中与 $H = p \cup \Delta_C$ 一致的样本集合 $\hat{D}_C$ 的大小，该集合可定义为如下形式：

$$\hat{D}_C = \arg\max_{D_c \subseteq D} \quad |D_c| \tag{5.3}$$
$$\text{s.t.} \quad \forall \langle \boldsymbol{x}_i, y_i \rangle \in D_c \quad (KB \cup \Delta_C \models p(\boldsymbol{x}_i) \wedge y_i).$$

式5.2中识别模型 p 的目标函数通常可微，因此一般可通过梯度下降、反向传播等连续优化手段进行训练；而知识模型 Δ_C 则是结构离散的逻辑公式，因此它的学习只能采用不可微的组合优化。如何将此二者纳入统一框架进行学习是反绎学习中最主要的挑战。

本文中，我们通过交替优化方法来进行反绎学习。即在固定识别模型 p 的情况下对 Δ_C 进行优化；而在固定知识模型 Δ_C 时对 p 进行优化。如此反复进行，直到 $H = p \cup \Delta_C$ 在训练数据 D 中达到最大一致。

注意到，当第 t 轮的识别模型 p^t 确定时，它在所有样本 D 中识别出的伪标记符号 $p^t(\boldsymbol{x}) = \cup_i p^t(\boldsymbol{x}_i)$ 能够与领域知识 KB 和所有样本标记 $Y = \{y_1, \ldots, y_n\}$ 一起作为知识精化的输入，最终获得满足 $KB \wedge p^t(X) \wedge \Delta_C^t \models Y$ 的 Δ_C^t。因此，我们可将 Δ_C^t 表示为 $\Delta_C(KB, p^t(X), Y)$。在 p^t 确定时，它可通过反绎推理得到。

然而当机器学习部件 p^t 未得到充分训练时，它无法输出准确的伪标记符号集。这种情况下，反绎逻辑推理很难得到与 KB 和 D 一致的 $\Delta_C(KB, p^t(X), Y)$。这时，我们便需要对伪标记 $p^t(X)$ 中可能出错的部分进行修改，以尽可能地令 Δ_C^t 与它们一致。我们记需要被修改伪标记符号为 $\delta(p^t(X)) \subseteq p^t(X)$，其中 δ 是一个用于估计哪些伪标记被识别错误的判别函数。

有了该判别函数，反绎逻辑推理便能根据领域知识 KB 修改 $\delta(p^t(X))$ 中标记出的错误符号，以得到更可能正确的伪标记符号集 $r_\delta(X)$。为了保证经过修改后的 $r_\delta(X)$ 不会偏离原来的伪标记 $p^t(X)$ 太多，我们还必须保证待修改符号 $\delta(p^t(X))$ 的个数尽可能少。

如此，我们便可将 p^t 固定时对 Δ_C 进行优化的问题转化为一个关于 δ 优化问题，式5.2中的目标被重新形式化为：

$$\begin{aligned} \max_{\delta} \quad & \mathrm{Con}(p^t \cup \Delta_C(B, r_\delta(X), Y) \cup D) \\ s.t. \quad & \mid \delta(p^t(X)) \mid \leq M \end{aligned} \tag{5.4}$$

其中常数 M 是允许 δ 对 $p^t(X)$ 进行修改的最大次数。该优化问题仍然十分困难。它本身是一个非凸问题，因此一般需要通过非梯度优化[175]方法解决。另外，为了提高学习效率，一般使用贪心搜索来解决式5.3中的子集选择（subset selection）优化问题。

得到 δ 的同时，只需通过反绎逻辑推理即可同时确定当前与 $KB \cup D$ 最大一致的知识模型 $\Delta_C^t = \Delta_C(KB, r_\delta(X), Y)$ 和被修正过的伪标记符号集 $r_\delta(X)$。

因此，确定知识模型 Δ_C^t 的同时，只需将 $r_\delta(X)$ 作为标记即可对第 $t+1$ 轮的识别模型进行训练：

$$p^{t+1} = \arg\min_{p} \quad \sum_{i=1}^{m} L(p(\boldsymbol{x}_i), r_\delta(\boldsymbol{x}_i)) \tag{5.5}$$

其中 L 为机器学习在训练 p 时使用的损失函数；$r_\delta(\boldsymbol{x}_i)$ 表示 KB 根据 δ 所识别的错误，在样本 $\boldsymbol{x}_i$ 上对 $\delta(p^t(\boldsymbol{x}_i))$ 使用反绎逻辑推理纠正后所得的新伪标记。

综上所述，反绎学习的过程大致如下：当输入数据时，首先由机器学习部件从数据中提取逻辑符号表示的伪标记 $p(X)$；这些伪标记被输入进反绎逻辑部件，用于增广领域知识并被用于知识推理与精化，以估计出样本的最终标记。当预测标记与真实标记出现不一致时，反绎学习会根据真实标记与领域知识反绎推理出样本中更可能正确的伪标记符号 $r_\delta(X)$，以重新训练机器学习模型。

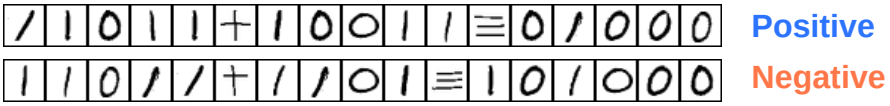


图 5.2 手写二进制加法等式学习的例子

5.2.3 基于反绎学习的 NLM 方法

为了验证反绎学习框架是否能够实现领域知识与机器学习的互促结合，本小节提出了 NLM（Neural Logical Machine）方法作为反绎学习的一种实现。此外，我们还设计了一类复杂的学习问题，以对 NLM 的效果进行检验。

二进制加法等式学习

为了通过实验验证反绎学习是否可行，本章中我们设计了一种对二进制加法等式进行学习的任务，图5.2中展示了该任务中所使用等式。每个等式由一个图片序列构成，且拥有一个标记，序列中的每个图片则对应于 $\{0, 1, +, =\}$ 中的某个符号。此外，该任务还包含了以下领域知识：

- 每个等式均符合“$X + Y = Z$”的结构；
- 等式中的运算是按位从右向左进行的，即 $abc + def$ 需要首先计算 $c + f$、$b + e$，最后才计算 $a + d$。

该任务的目标是学会预测未见图片序列中的等式是否正确。需要注意的是，该任务中并未指定 *加法类型*。因此数据中的等式可能满足二进制算数加法，即 $0 + 0 = 0$、$1 + 0 = 1$、$0 + 1 = 1$ 和 $1 + 1 = 10$；也有可能是按位逻辑运算，例如逻辑异或：$0 + 0 = 0$、$1 + 0 = 0$、$0 + 1 = 0$ 和 $1 + 1 = 1$。

根据定义5.1，该问题中的目标概念 C 为“正确等式”，样本 D 为特征空间 $\mathbb{R}^d$ 中的等式图片以及它们的标记，领域知识 KB 则是等式结构和按位运算，需要识别的基本概念 $\mathcal{P} = \{0, 1, +, =\}$。待学习的识别模型 $p : \mathbb{R}^d \mapsto \mathcal{P}$ 是一个从等式图片中提取数学符号的映射，而知识模型 Δ_C 则是用于定义加法类型的逻辑公式。

NLM 的结构与工作流程

NLM 主要由三个主要模块构成，它们分别对应于反绎学习框架的三个部件，图5.3 中展示了它的结构。最上层是由卷积神经网络（convolutional neural network，CNN）[95] 构成的识别模型，它对应着反绎学习中的机器学习部分；

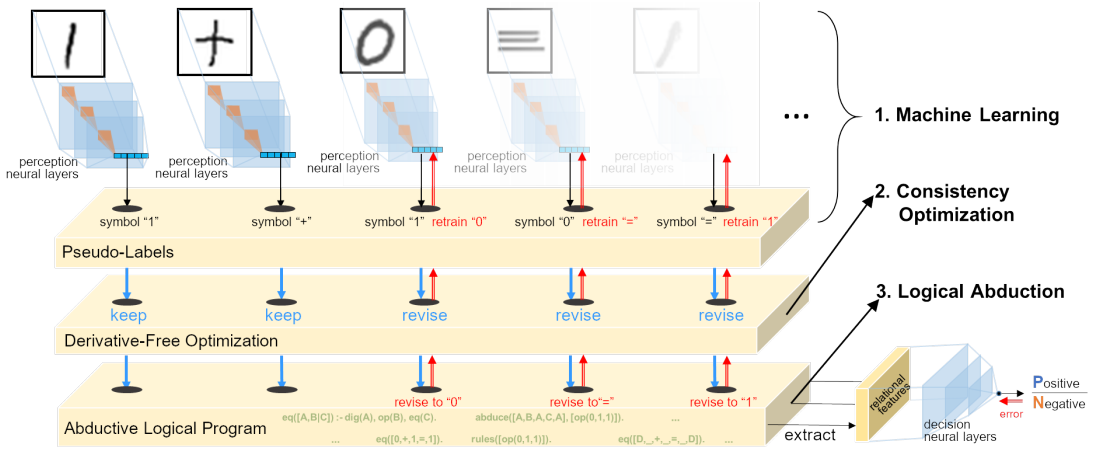


图 5.3 NLM 方法

下部的反绎逻辑程序（Abductive Logic Programming，ALP）[73] 则对应于反绎逻辑推理部件；中间是则是由非梯度优化方法实现的一致性搜索[175]。在手写等式加法学习任务中，CNN 的目标是学会一个从图片中识别等式符号的模型；ALP 的任务则是根据领域知识和样本标记进行知识精化和推理，既需要根据 CNN 的识别的等式符号学会数据中的加法形式（例如二进制算数加法、逻辑与、逻辑异或等等），又需要在识别结果与领域知识和样本标记不一致的情况下推理出更正确的等式符号；而非梯度优化则在该过程中最大化学得模型与领域知识、训练样本的一致性。

当训练开始时，经过初始化的 CNN 会首先尝试将手写等式图片识别为“0”、“1”、“+”和“=”符号，并根据图片顺序将它们作为等式输入进 ALP 中。一开始，随机初始化的 CNN 识别精度很低，因此识别所得的等式多半为错误的，因此 ALP 很难精化出与样本标记和领域知识一致的 Δ_C。也就是说，根据 CNN 识别的等式无法搜索到任何加法规则使得所有等式的正确性符合均其标记描述。因此 NLM 需要对 CNN 输出的符号 $p(X)$ 进行修改，并对它进行重新训练以提高符号识别精度。

ALP 能够根据领域知识判断识别出来的等式是否和样本标记一致，但它在发现不一致的情况下却无法得知具体是哪些符号中的识别错误导致了不一致。根据5.4式中描述的目标，我们需要学习一个用于猜测错误识别位置的函数 δ，该函数的形式可能非常复杂。在实现中，NLM 通过非梯度优化方法 RACOS[175] 来学习该函数。当给定一个 δ 时，它可以标记出等式中大概率出现识别错误的位置，然后由 ALP 借助领域知识和样本标记来尝试对这些错误位置进行修改，并同时给出修改时所依据的 Δ_C。RACOS 则在优化过程中不断地搜

算法 5.1 Neural Logical Machine (NLM)

算法输入：训练样本 D；领域知识 KB；最大迭代次数 T；最小一致样本数 $Threshold$
算法过程：
// 初始化 *CNN* 网络 p 和知识模型（*NLM* 的关系型特征集合）Δ_C
$t = 0$
$p^t = \text{Init}_{\text{CNN}}(p)$
$\Delta_C = \Phi$
for $t < T$ **do**
  // 对样本进行随机采样
  $D^t = \text{Sample}(D)$
  // 当假设模型 $H^t = p^t \cup \Delta_C^t$ 未达到终止条件时同时学习 p 和 Δ_C^t
  while $Con(H^t \cup D^t) < Threshold$ **do**
    对式5.9进行一致性优化，得到用于标记伪标记错误的向量 $\boldsymbol{g}$
    使用 $\boldsymbol{g}$ 对伪标记进行修改并进行反绎推理，分别得到 $r_\delta(X^t)$ 和 Δ_C^t
    优化式5.5得到 p^{t+1}
  end while
  // 将与采样 D_t 一致的 Δ_C^t 保存为关系型特征
  $\Delta_C = \Delta_C \cup \{\Delta_C^t\}$
end for
// 使用 Δ_C 将训练样本命题化为一个属性-值数据集 $U = \{\langle \boldsymbol{u}_i, y_i \rangle\}$
$U = \text{Propositionalize}(D, \Delta_C)$
// 使用命题化后的数据训练决策模型 h
$h = MLP(U)$
算法输出： $H = p \cup \Delta_C \cup h$

索最优的 δ^*，以使 ALP 输出的 Δ_C^* 与尽可能多的训练样本一致。同时，这些基于 δ^* 的修改结果将被用于重新训练 CNN 识别模型。在 NLM 的实现中，$\delta*$ 的输出为一个布尔向量 $\boldsymbol{g}$，它标记了 $p(X)$ 中需要被修改的符号位置。

由于 ALP 基于离散搜索的推理效率较低，通过反绎逻辑推理得到一个与所有训练样本均一致的 $\Delta_C = \Delta_C(B, p(X), Y)$ 十分困难。因此，NLM 在每轮学习时会从 D 中采样一部分训练数据进行反绎学习。这种做法虽然在第 t 轮学得的 Δ_C^t 能够对采样 $D^t = (X^t, Y^t)$ 一致，但很可能对于全体样本 E 不完备。例如，某轮采样得到的等式均为“0+1=1”、“10+1=11”和“100+11=111”这样的图片，知识精化所得的 Δ_C^t 只需包含 $0 + 1 = 1$ 这一条规则即可满足当前采样中的所有样本，但它却不能完整地定义二进制加法，所以不能被用于对未见等式进行预测。

因此，在进行最终决策前，NLM 会将每次采样后习得的 $\{\Delta_C^t\}$ 作为特征保留下来。当 NLM 达到终止条件时，它会采用类似第2章中 SUL 的方式将所有训练样本 D 通过特征 $\{\Delta_C^t\}$ 进行增广，将每个样本重新表示为一个关于 $\{\Delta_C^t\}$ 的布尔向量。最终以这些向量为特征学习一个全连接神经网络来作为决策层，以

```
% predicate for parsing symbol list
parse_eq(X, Y, Z, Eq) :- phrase(eq(X, Y, Z), Eq).

% definition of "digits" and "equation"
digit(0).
digit(1).
digits([D]) --> [D], { digit(D) }.
digits([D | T]) --> [D], !, digits(T), { digit(D) }.
eq(X, Y, Z) --> digits(X), [+], digits(Y), [=], digits(Z).
digits(X) :- phrase(digits(X), X).
```

(a) 关于等式结构的领域知识

```
% rule for calculation when bitwise addition rules available
calc(X, Y, Z) :- op_rules(R), bitwise_calc(R, X, Y, Z).

% bitwise calculation that handles carrying
bitwise_calc(R, X, Y, Z) :-
    reverse(X, X1), reverse(Y, Y1), reverse(Z, Z1),
    bitwise_calc_r(R, X1, Y1, Z1), !.
bitwise_calc_r(_, [], Y, Y).
bitwise_calc_r(_, X, [], X).
bitwise_calc_r(R, [D1 | X], [D2 | Y], [D3 | Z]) :-
    Op_Rule = my_op(D1, D2, Sum),
    member(Op_Rule, R),
    ((Sum = [D3], Carry = []); (Sum = [C, D3], Carry = [C])),
    bitwise_calc_r(R, X, Carry, X_carried),
    bitwise_calc_r(R, X_carried, Y, Z).
```

(b) 关于按位运算的领域知识

图 5.4 手写二进制等式加法学习任务中使用的领域知识

减小随机采样带来噪声对反绎学习的影响。此外，在手写二进制加法等式学习任务中，NLM 还采用了课程学习[9] 策略进行训练，即最从短等式学起，在学习过程中逐渐增加训练等式长度。下面几小节中将对 NLM 用到的领域知识、一致性优化方法和决策层进行详细介绍。

领域知识与反绎逻辑推理

本小节中将结合手写二进制加法等式任务中的领域知识对 NLM 反绎逻辑推理过程进行介绍。

图5.4中展示了 NLM 使用的领域知识 KB，它是由一阶逻辑公式构成的逻辑程序，由 Prolog 语言实现。其中图5.4(a)为关于样本中等式结构的知识，图5.4(b) 为关于按位运算（bitwise operations）的领域知识。图中蓝色与绿色关键字分别为 KB 定义的与标准 Prolog 谓词，黄色关键字为可反绎谓词。

图5.4(a)中的 Prolog 程序主要负责将 CNN 识别的符号序列解析为加法等式。NLM 的输入样本为 $\boldsymbol{x}_i = [I_{i1}, I_{i1}, \ldots, I_{ik}]$，其中每个 $I_{ij} \in \mathbb{R}^d$ 是一张手写字符图片。CNN 模型 p 会将它们识别为一个伪标记符号序列 $p(\boldsymbol{x}_i) = [s_{i1}, \ldots, s_{ik}]$，其中 $s_{ij} \in \{0, 1, +, =\}$。

随后，这些伪标记会被作为基本概念符号 $Eq = p(\boldsymbol{x}_i)$ 输入 ALP，而后者会调用图5.4(a)第 1 行中的谓词 $parse_eq/4$ 将其解析为 $X + Y = Z$ 的形式。解析过程会用到第 5-10 行中对等式结构的定义，其中：

$$eq(X, Y, Z) --> digits(\boldsymbol{x}), [+], digits(Y), [=], digits(Z). \tag{5.6}$$

是一条 Prolog 中的 DCG（Definite Clause Grammars）规则[14]，它可以将 $p(\boldsymbol{x}_i) = [1, +, 1, 0, 1, =, 1, 1, 0]$ 这样的序列解析成原子公式 $eq([1], [1, 0, 1], [1, 1, 0])$。它所调用的第 5-8 和第 10 行中的规则以递归的形式定义了数字字符（digit）和数字（digits）的概念：每个数字是由 $\{0, 1\}$ 构成的序列，例如 [0] 和 $[1, 0, 0, 1]$ 等。

图5.4(b)中的规则负责对 $parse_eq/4$ 解析而得的 $eq(X, Y, Z)$ 进行按位运算，实现该操作的是下面规则定义的谓词 $calc(X, Y, Z)$：

$$calc(X, Y, Z) \leftarrow op_rules(R), bitwise_calc(R, X, Y, Z). \tag{5.7}$$

其中 $op_rules/1$ 被用于定义一组未知的按位运算规则 R；而 $bitwise_calc(R, X, Y, Z)$ 则根据规则集 R 将 X、Y、Z 倒排（第 6 行），然后按位对 $X + Y$ 进行运算以得到 Z（第 8-15 行）。

R 中的规则即是 ALP 中的可反绎事实，也就是待学习的知识模型 Δ_C，其中的每条规则由谓词 $my_op(A, B, C)$ 表示。该谓词定义的是按位运算法则 $A + B = C$，其中 A 和 B 均为单比特的数字字符，而 C 则是一般的数字。例如，逻辑异或（xor）即可通过

$$R = [my_op(0, 0, [0]), my_op(1, 0, [1]), my_op(0, 1, 1), my_op(1, 1, [0])] \tag{5.8}$$

进行定义。此外，第 5-15 行中对 $bitwise_calc/3$ 的实现允许进位操作，例如二进制算数加法中的 $my_op(1, 1, [1, 0])$。如前文所说，NLM 使用的领域知识中没有对 R 中的运算进行定义，因此它必须通过反绎逻辑推理而得到。

式5.6中的 $eq(X, Y, Z)$ 同样属于可反绎谓词，这是因为当 $p(X) = [s_1, s_2, \ldots, s_k]$ 中有符号出现错误时，ALP 还必须根据领域知识对其进行纠正。比如对

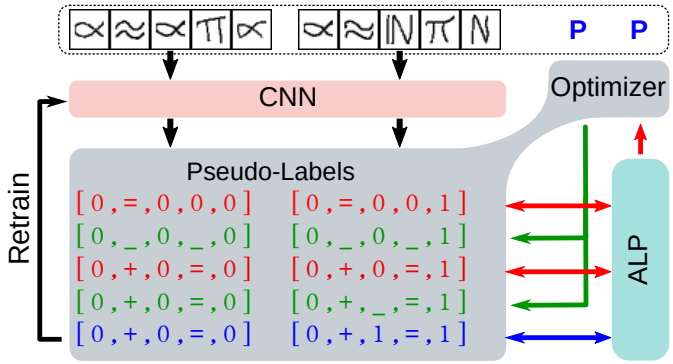


图 5.5 反绎推理对 $p(\boldsymbol{x})$ 的修改与 CNN 的重训

于一个正样本 $\langle \boldsymbol{x}_i, y_i \rangle$ 且 $y_i = true$，若初始化的 p^0 输出的伪标记结果为 $Eq_0 = [1,1,1,1,1]$。那么，$parse_eq/4$ 便无法将其解析为等式 $X + Y = Z$，因为 Eq_0 中加号和等号都不存在，因此无法反绎推理出合法的运算法则 $\Delta_C = R$，并导致 $bitwise_calc(R,X,Y,Z)$ 运算失败，最终 ALP 将返回 $p^0(\boldsymbol{x}_i)$ 和 $y_i = true$ 不一致。

这时，便需要依靠一致性优化过程中学得的 δ 来判断 $p^0(\boldsymbol{x}_i) = Eq_0$ 中的哪些位置出现错误。δ 可以将错误的 Eq_0 中错误的位置替换为 Prolog 变量，例如当 δ 发现无论怎样替换第 1、3、5 位都不可能使其与其标记一致时，它终将尝试出将第 2 和第 4 位标记为待修改位置。那么，Eq_0 会被修改为 $Eq_1 = [1,_,1,_1]$，其中“_”为 Prolog 语言中用于表示“未知”的空缺变量。然后，ALP 会尝试对 $eq(X,Y,Z)$ 和 R 同时进行反绎推理，以使得 $r_\delta(\boldsymbol{x}_i) = Eq'_1$ 和 $\Delta_C = R$ 能够满足 $KB \cup r_\delta(\boldsymbol{x}) \cup \Delta_C \models y$。其中一个简单的解答便是令 $Eq'_1 = [1,+,1,=,1]$ 且 $R = \{my_op(1,1,[1])\}$。最后，被反绎推理得到的 $r_\delta(\boldsymbol{x}_i) = Eq'_1$ 将被用于重新训练 CNN，令其逐渐学会区别“+”、“=”与其他数字。而 $\Delta_C = R$ 则会被用于检验在其他样本中的一致性，并根据结果而被排除或接受。

图5.5中展示了以上反绎逻辑推理的过程，其中红色箭头表示 ALP 判断出 $p(X)$ 与领域知识、样本标记不一致，绿色箭头表示一致性优化（optimizer）得到的 δ 对可能错误的位进行标记并将其修改为空缺变量，蓝色箭头表示 ALP 通过反绎推理最终得到了一致的 $r_\delta(X)$ 和 Δ_C，其中前者被用于重新训练 CNN 模型以提升其识别精度。

NLM 中的反绎逻辑推理与第3、4章中相同，它所做的推理依然还是根据领域知识 KB 对已观测事实 O 推理出一种解释 Δ。在 NLM 中 O 为从手写二进制等式图片识别的伪标记 $p(X)$ 以及等式标记 Y；Δ 则包括加法准则 $\Delta_C = R$ 以

及被修改的伪标记符号 $r_\delta(X) = Eq'_1$。反绎推理的过程中同样可以加入完整性限制规则 IC，例如 Δ_C 中不能同时拥有冲突规则 $my_op(1,0,[1])$ 和 $my_op(1,0,[0])$ 等等。

理想情况下，已观测事实 O 应当包含所有训练数据 D。但是在一阶逻辑中对公式的一致性检测是一个 NP-难问题，因此将样本全部用于 ALP 推理并不可行。在5.3节中的实验里，NLM 的每轮学习仅采样 5-10 个等式用于反绎逻辑推理。每次采样中学得的运算法则被保存为关系型特征用于对 D 进行命题化[90]后训练一个决策模型。

一致性优化

上一小节中提到，当 NLM 的识别模型 p 未经过充分训练时其输出的等式符号序列 $p(\boldsymbol{x}_i) = Eq$ 很可能出现错误，因而需要通过 ALP 来对 $p(\boldsymbol{x}_i)$ 序列进行修改以得到用于重训 CNN 的 $r_\delta(\boldsymbol{x}_i)$。为了完成这一过程，NLM 需要使用一致性优化来获得 δ 并为 ALP 标记出应当修改的比特 $\delta(p(\boldsymbol{x}_i))$。

NLM 在实现中将 $\delta(p(\boldsymbol{x}_i))$ 表示为一个布尔向量 $\boldsymbol{g}_i = (g_1, \ldots, g_k)$，其中 k 为 $p(\boldsymbol{x}_i)$ 序列的长度，$g_j = 1$ 表示该位置需要被改为 Prolog 空白变量“_”以供 ALP 进行修改。那么，我们便可以通过 g 将 $r_\delta(\boldsymbol{x}_i)$ 重新写为 $r(\boldsymbol{x}_i, \boldsymbol{g})$。而当输入为采样得到的多个等式图片序列集合 D^t 时，它可以表示为 $r(X^t, \boldsymbol{g})$。其中，我们记 X^t 为 D^t 中所有样本的特征向量，Y^t 为所有标记。这时 D^t 中被修改伪标记个数即为 $\boldsymbol{g}$ 中非零元的个数。最终，式5.4中对 δ 的优化被改写为：

$$\begin{aligned} \max_{\boldsymbol{g}\in\{0,1\}^k} \quad & \mathrm{Con}((p^t, \Delta_C(B, r(X^t, \boldsymbol{g}), Y^t)) \cup D^t) \\ \text{s.t.} \quad & |\boldsymbol{g}|_0 \le M \end{aligned} \tag{5.9}$$

该式表示的布尔变量优化问题假设空间可能十分复杂，传统基于梯度的连续优化方法难以适用。NLM 采用非梯度优化方法 RACOS 来对该问题进行求解[175]。RACOS 是一类随机优化方法，它通过学习一个关于解空间的分类器来评估每个随机生成解的质量，它在许多复杂的优化问题中取得了较好的效果[175]。直观来看，在式 5.9的问题中，RACOS 学习的分类器事实上就是用于评估等式中哪些位置最容易出错的函数 δ。

而在计算样本集 D^t 的最大一致样本数时需要对式5.3进行优化，这是一个 NP-难的子集选择问题，因此 NLM 选择使用贪心搜索来近似求解。

决策学习

前文提到，仅使用部分训练数据 D^t 时，ALP 的反绎推理得到的知识模型 Δ_C^t 很可能不完备。此外，当 p^t 未充分训练时，ALP 基于 $p^t(X)$ 反绎推理所得的 Δ_C^t 也很有可能不正确。所以它很难被用于进行最终决策。

受 ILP 中命题化[90] 方法的启发，NLM 选择将每一轮反绎推理所得的运算法则 Δ_C^t 作为关系型特征。当 CNN 模型 p 收敛后，NLM 会从中选择多个与训练集中正样本最一致的 Δ_C^t 作为特征。最终，所有训练样本 $\{\langle \boldsymbol{x}_i, y_i\rangle\}$ 将被这些特征增广为一个特征向量。

命题化数据集的生成过程如下：对训练样本 $\langle \boldsymbol{x}_i, y_i\rangle$，首先使用已收敛的 CNN 识别模型 p 将其转化为等式符号序列 $p(\boldsymbol{x}_i)$，然后将利用 N 个与正样例最一致的关系型特征将 $p(\boldsymbol{x}_i)$ 转化为布尔向量 $\boldsymbol{u}_i = [u_{i1}, \ldots, u_{iN}]$，其中：

$$u_{ij} = \begin{cases} 1, & KB \cup p(\boldsymbol{x}_i) \cup \Delta_C^j \models y_i, \\ 0, & KB \cup p(\boldsymbol{x}_i) \cup \Delta_C^j \not\models y_i. \end{cases} \tag{5.10}$$

通过该方法，数据集 D 被转化为一个属性-值数据集 $U = \{\langle \boldsymbol{u}_i, y_i\rangle\}$。最终，NLM 从该数据集中学习一个多层感知机（multilayer perceptron，MLP）作为决策模型。当 NLM 进行预测时，测试样本也经过同样的步骤被转化为关于关系型特征的特征向量，并通过决策模型判断其标记。

一方面，关系型特征中的 Δ_C^t 形式为一阶逻辑公式，它们依然拥有良好的可理解性，且能够作为知识模型进行逻辑推理。另一方面，通过引入命题化方法与决策模型，NLM 能够降低学习过程随机采样所引入噪声的影响。

5.3 实验验证

本节将在上一节中介绍的二进制等式加法任务中对基于反绎学习的 NLM 算法进行实验验证。我们一共进行了两类实验，第一种为二进制加法等式图片分类实验，第二种为模型迁移实验。此外，我们详细地记录了实验的中间结果，以验证反绎学习框架是否能够实现机器学习模型与领域知识的共同提升和精化。下面的小节中，我们将分别对实验数据和两类实验进行介绍。

实验数据

本节中，我们构造了两类手写二进制加法等式数据集以验证 NLM 是否有能力实现领域知识与机器学习的互促结合。数据的例子请见图5.6，它们分别来自于两种手写字符数据集：

- **手写数字二进制加法等式（Digital Binary Additive，DBA）**：该数据集中等式数字的图片均来自于手写数字图片数据集 MNIST[95]，加号与等号的图片则来自于 HASYv2 数据集[158]。
- **随机字符二进制加法等式（Random Binary Additive，RBA）**：该数据集中的数字、加号与等号均随机采样自多语言字符数据集 Omniglot[92]。

两个数据集中的等式数字和符号均一一对应，区别仅在于每个符号所使用的图片。为了验证所有对比方法在未见等式上的泛化能力，本节中所有实验的训练集与测试集均不相交。数据的输入形式均为像素表示的原始图片序列。NLM 所引入领域知识仅包括等式结构和按位运算算法，加法的具体运算法则需要从数据中习得。关于它们的详细介绍请见第5.2.3小节以及图5.4。

数据集中的等式均由 $\{0, 1, +, =\}$ 共 4 种符号构成，其中的等式最短长度为 5，最大为 26。它们被分为两类，其中正样本图片序列为计算正确的等式，负样本则反之。此外，我们为这两个数据集构造了两类加法任务，一类是普通的二进制算数加法（add），另一类为逻辑异或（xor)。

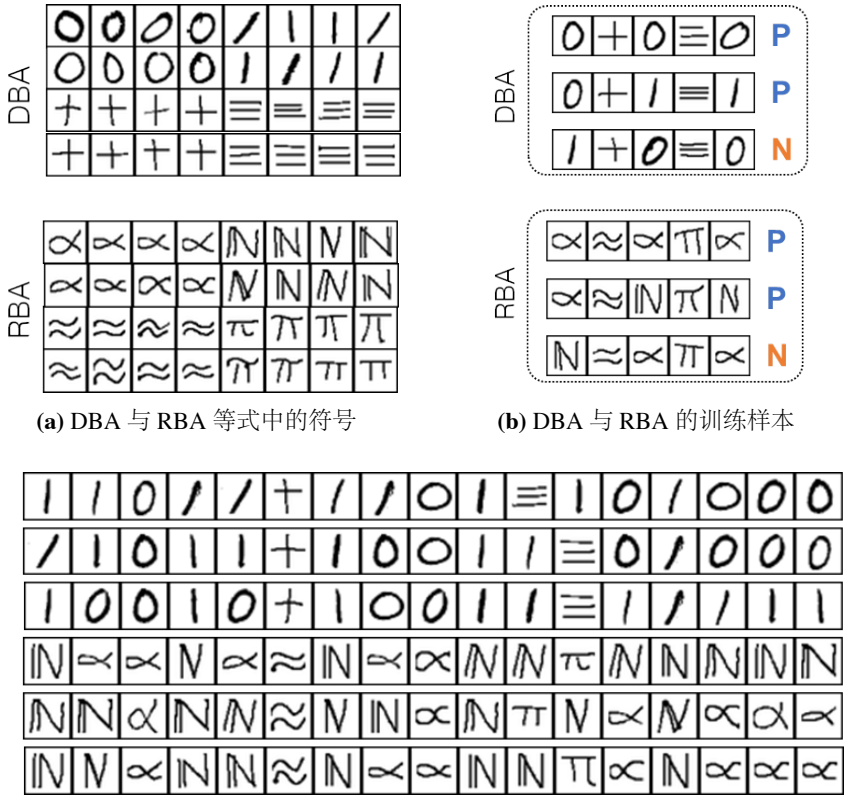
本节设计了两种实验：1) 二进制加法等式学习，它被用于检验 NLM 是否能够从数据中学得识别模型和知识模型；2) 迁移学习，它被用于检验 NLM 习得的识别模型和知识模型是否能够在不同的任务之间迁移。

5.3.1 实验 1：二进制加法等式学习

实验设置

该实验的目标是对 DBA 以及 RBA 数据集中的二进制算数加法等式进行学习，并在未见过的等式中进行预测。本实验中分别对比了以下几种方法：

- **NLM-all**：这是本文中提出的方法。它的结构如图5.3所示，其中实现机器学习部件的识别模型为一个 2 隐层 CNN，决策层为一个 2 隐层 MLP，它最终会保留 50 个 Δ_C^t 作为关系型特征，其他部分均如5.2.3小节所介绍。该方法使用 5-26 位的等式作为训练集，其中每个长度 300 个，共 6600 个。



(a) DBA 与 RBA 等式中的符号

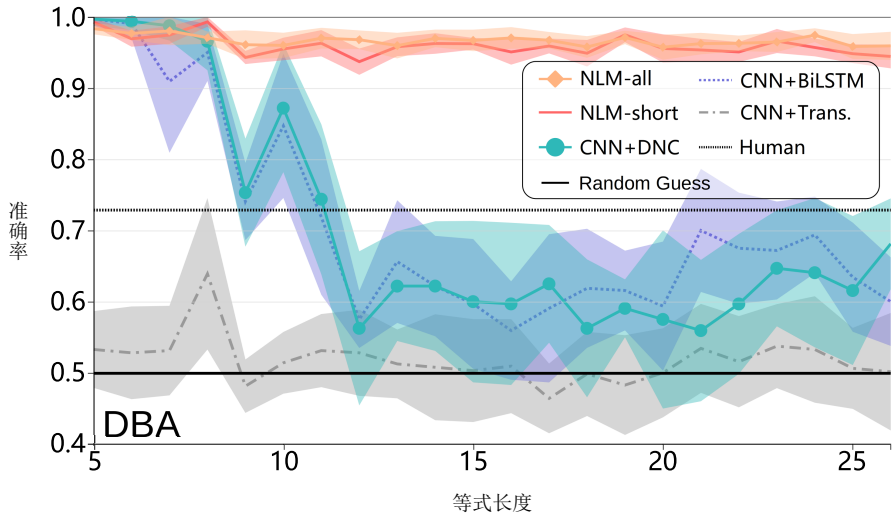
(b) DBA 与 RBA 的训练样本

(c) DBA 与 RBA 的测试样本

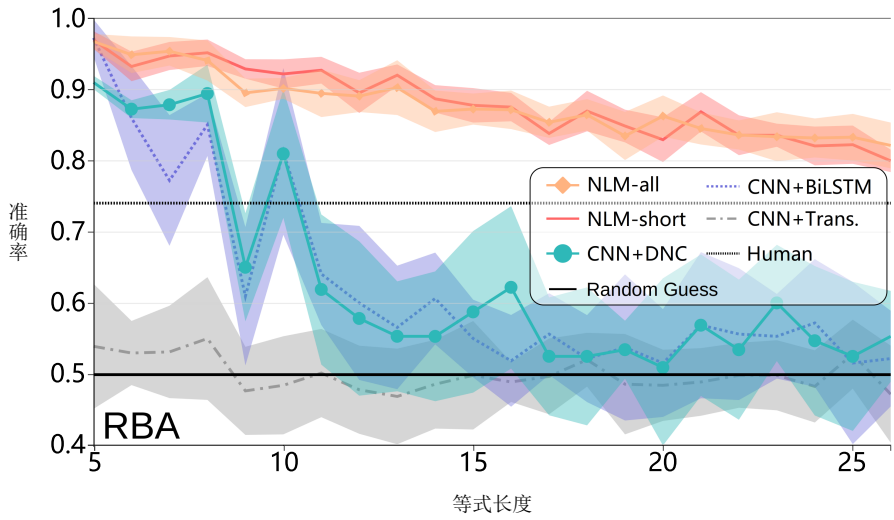
图 5.6 DBA 与 RBA 数据集的例子

- **NLM-short**：该方法的结构和参数与 NLM-long 相同，但它只使用长度为 5-8 的等式训练，其中每个长度等式有 300 个，共计 1200 个训练样本。
- **DNC（Differentiable Neural Computer）**[58]：该方法是一种基于外部存储（memory）的 DNN，因此它能够处理需要一定推理能力的机器学习问题，甚至在一些较复杂的符号推理问题中也能取得不错的效果[58]。该方法使用的训练数据与 NLM-all 相同。
- **Transformer**[163]：该方法是近年来最好的序列学习（sequential learning）DNN 之一，它通过注意力（attention）机制进行学习，在许多自然语言处理任务中得到了成功应用。该方法使用的训练数据与 NLM-all 相同。
- **BiLSTM（Bidirectional Long Short-Term Memory Network）**[150]：该方法是最经典的序列学习神经网络方法之一，被成功应用在大量机器学习任务中。该方法使用的训练数据与 NLM-all 相同。本节实验中将其作为基准方法（baseline）进行对比。

所有方法中的神经网络模型均根据它们在预留验证集上的训练精度进行



(a) DBA 算数加法等式任务中的测试精度



(b) RBA 算数加法等式任务中的测试精度

图 5.7 手写二进制加法等式学习实验中的测试精度

调参。为了验证对比方法的推理能力，DNC、Transformer 和 BiLSTM 均采用与 NLM 相同的 2 隐层 CNN 作为等式图片的输入层。其中 Transformer 需要在 CNN 后增加一个嵌入式编码（embedding）层，该层的参数同样通过验证集进行选取。为保证实验结果可靠，每个实验均重复 10 次。由于问题较为复杂，所有模型均采用课程学习[9] 策略进行训练，其中第 n 个课程即为长度为 n 训练样本。

此外，为了研究人类如何解决此类复杂学习问题，本节还进行了一部分人类实验，共 40 个志愿者参与测试。测试开始之前，所有被试者均会被告知与

```
1 % symbolic hypothesis abduced from arithmetic addition tasks.
2 R_art = [my_op(0, 0, [0]), my_op(1, 0, [1]),
3          my_op(0, 1, [1]), my_op(1, 1, [1, 0])].
4
5 % symbolic hypothesis abduced from exclusive-or tasks.
6 R_xor = [my_op(0, 0, [0]), my_op(1, 0, [1]),
7          my_op(0, 1, [1]), my_op(1, 1, [0])].
```

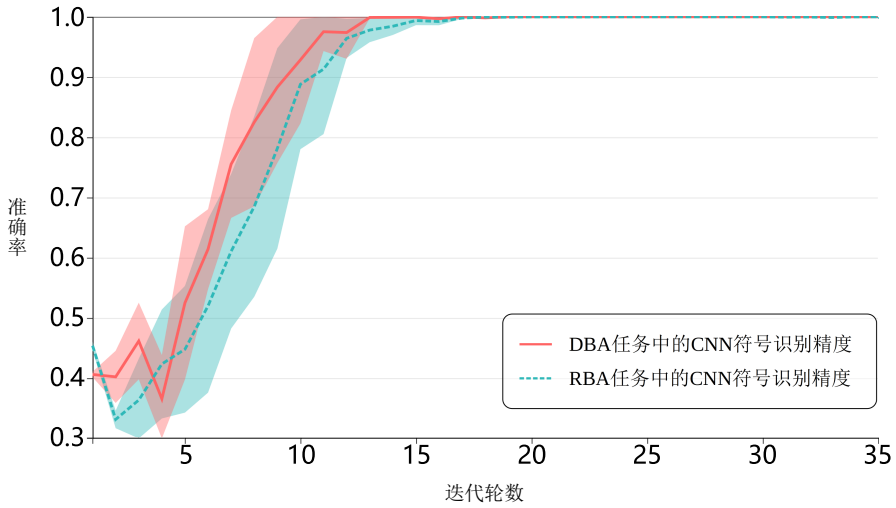
图 5.8 NLM 学得的知识模型

NLM 同样的领域知识。人类测试的设置与以上实验稍有不同。首先，训练集中只包含 5 个正样本与 5 个负样本，测试集共有 5 个样本，所有等式的长度均在 5-14 之间；其次，由于二进制算数加法对于人来说比较简单，因此该实验中的等式均根据随机生成的二进制运算构造。随机生成的运算中不存在 $a+b$ 等于多个结果的冲突法则，且满足交换律；最终，每次测试（包括观察训练样本）不超过 6 分钟，被试者会根据预测对的数量被给予奖励。尽管该设置下的人类实验与机器学习任务没有直接的可比性，但它或许能在一定程度上帮助我们观察人类思维与机器学习的异同。

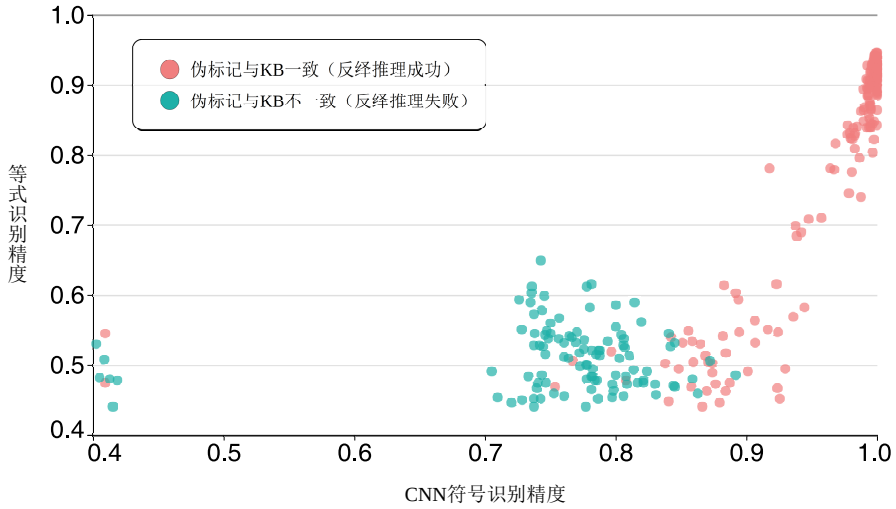
实验结果

图5.7(a)和图5.7(b)分别展示了所有对比方法（包括人类实验）在 DBA 和 RBA 任务中的结果，其中纵轴为测试精度，横轴为测试等式的长度，阴影表示多次实验所得结果的标准差。可以看到，NLM 方法的泛化性能显著优于其他神经网络方法，但 NLM-all 和 NLM-short 的表现在两个任务中没有显著区别。此外，所有方法在 DBA 任务中的表现均超过了它们在 RBA 任务中的表现，这是因为 DBA 中使用的符号图片更容易区别。对比方法中，BiLSTM、DNC 均随着测试等式长度增长而迅速下降，Transformer 的效果与随机猜测相近。NLM 方法的精度下降速度显著低于其他方法，且基本维持在 80% 以上。

除了分类精度，我们还研究了 NLM 的知识精化效果。图5.8中的第 1-3 行展示了 DBA 和 RBA 二进制加法等式学习任务中与训练样本最一致的 Δ_C^t。可以看到，它的确学会了二进制算数加法的运算法则。事实上，决策层最终保留的大部分 Δ_C^t 均准确地学习出了图5.8中的运算法则，只有小部分缺少 1 到 2 条法则。由于学得了准确的知识模型，所以即便训练等式长度只有 5-8 的 NLM-Short 也在长度最多为 26 的测试等式上取得了很好的泛化效果。



(a) 训练过程中识别模型的符号识别精度变化



(b) 训练过程中一致性优化结果与识别模型精度关系

图 5.9 NLM 在训练过程中的中间结果

为了检验 NLM 是否能够在学习过程中使得领域知识精化和机器学习模型相互促进，我们还记录了训练过程中识别模型 p 的训练精度以及 ALP 反绎逻辑推理的结果，它们分别被展示在图 5.9(a)图和5.9(b)中。可以看到，尽管训练数据中不存在对识别模型 p 的直接监督信息，但图5.9(a)显示它的训练精度仍然能够随着迭代轮数的增加而增长。该现象验证了基于领域知识的反绎逻辑推理的确可以为机器学习提供正确的监督信息，纠正识别模型中错误的识别结果并促进它的学习。图5.9(b)展示了 NLM 训练过程中一致性优化结果与模型精度的关系。每个点代表一次反绎逻辑推理，点为红色表示此次反绎推理找到了与领

域知识以及样本标记一致的知识模型 Δ_C，绿色则表示未找到——即 NLM 需要对 CNN 识别的伪标记 $p(X)$ 进行修改。纵轴表示模型当前状态下判断等式正误的训练精度（equation accuracy），横轴表示模型当前状态下符号识别模型的符号识别精度（perception accuracy）。可以看到，当符号识别精度超过 80% 后，大部分的反绎逻辑推理均能够找到与领域知识和样本标记一致的知识模型。此外，随着 CNN 的符号识别精度上涨，等式判别精度也会逐渐增加。该结果验证了 NLM 确实能够通过机器学习促进知识精化。不过该结果也表明，只有当符号识别模型足够精确时才能帮助知识精化。

人类实验的结果却显示了与 NLM 完全相反的情况：被试者均表示在符号识别上并未遇到任何困难，而知识精化才是更复杂的问题。这与 Simon 和 Newell 在 1970 年得出的结论一致[153]。由于人类实验中的加法运算法则更复杂，因此在 DBA 和 RBA 上分别只取得了 72.86% 和 74.00% 的精度。可以看到，不同符号图像的差别并未对测试结果造成影响。因此，短时间内进行精确的逻辑推理和知识精化才是 NLM 与人类相比的优势弱项。

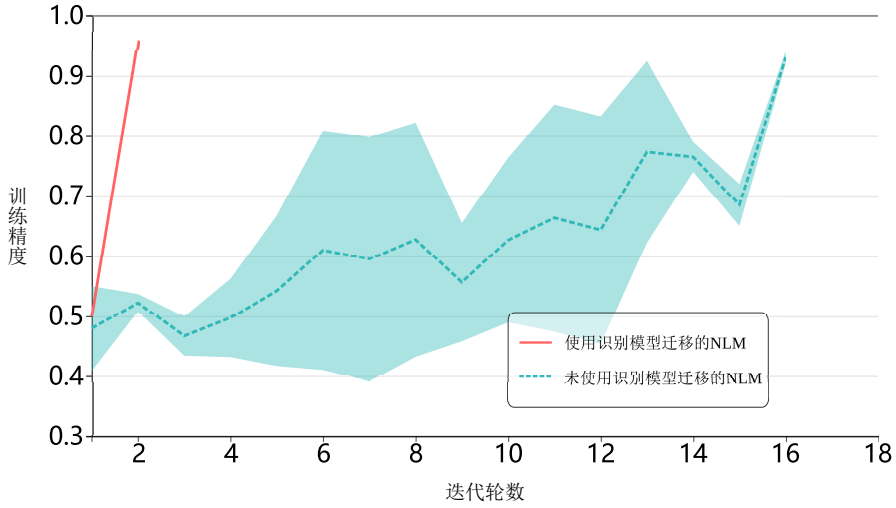
5.3.2 实验 2：迁移学习

实验设置

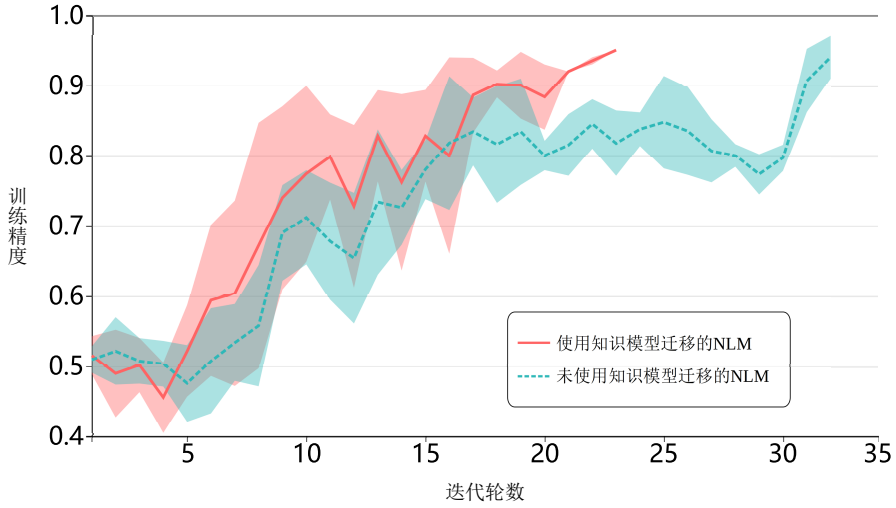
该任务的目标是分别检验 NLM 学得的识别模型和知识模型是否能够在不同类型的任务中进行迁移。

第一个任务是识别模型迁移（perception transfer）。为了完成该实验，我们构建了一个知识模型结构与二进制加法截然不同的等式数据集——逻辑异或（xor）等式数据集，其中所有的等式图片序列均由 DBA 数据生成，其大小与 NLM-all 所使用的 DBA 二进制加法等式数据集相同。为了将 DBA 二进制加法等式学习得到的识别模型迁移至异或等式领域，我们将前者任务中学得的模型参数和结构全部固定，因此 NLM 只需只进行反绎逻辑推理以得到 Δ_C^t，然后对训练样本进行命题化来学习最后的决策层。并以另一个采用相同训练数据、但从零开始学习的 NLM 作为对比。

第二个任务为知识模型迁移（knowledge transfer）。该实验中，我们将 RBA 二进制加法等式学习任务中学得的知识模型迁移至 DBA 二进制等式加法。被迁移的知识模型包括所有用于命题化的关系型特征 Δ_C^t 以及最后的决策



(a) 识别模型 p 迁移的训练精度



(b) 知识模型 Δ_C 迁移的训练精度

图 5.10 NLM 迁移学习任务中的训练精度

层参数。因此 NLM 只需学习一个 CNN 识别模型。作为对比，另一个 NLM 需要从零开始在 DBA 二进制等式加法数据集中进行学习。

实验结果

图5.10(a)展示了识别模型迁移的结果，它学得与正样本最一致的 Δ_C^t 请见图5.8的第 5-7 行。从结果中可以看出，进行了识别模型迁移的 NLM 与从零开始学习的 NLM 在最终训练精度上相差不大，但进行了识别模型迁移的 NLM 能够迅速收敛。该结果与图5.9(b)中的现象一致，即只要识别模型精度足够高，反绎逻辑推理便更容易进行知识精化。

从图5.10(b)中的结果可以看出，两个 NLM 最终能够达到相似的训练精度；但是进行了知识迁移的 NLM 收敛速度比从零开始的 NLM 更快。这说明正确的领域知识能够加速识别模型的机器学习。不过，在学习的前半段二者的表现却较为相似。此外，通过横向对比 5.10(a)与图5.10(b)中的识别模型迁移与知识模型迁移结果可以再次发现，当识别模型未充分训练时带来的困难更大。

5.4 小结

本章考虑了在结合领域知识与机器学习时如何令二者相互促进、共同学习。在一些较复杂的学习问题中，我们没有能够直接用于训练机器学习模型的监督信息，只能根据领域知识通过反绎逻辑推理而得到。另一方面，基于领域知识的逻辑推理却又必须以机器学习模型输出的符号作为基本事实。此二者相互依赖，所以该类问题比常规机器学习问题更加困难。

为尝试解决此问题，本章中提出了反绎学习框架，试图通过一致性优化将机器学习与基于领域知识的反绎逻辑推理相结合，以达到知识精化和机器学习共同进行、相互促进的目的。基于反绎逻辑框架，本章还提出了 NLM 方法，并将其应用于二进制加法等式学习任务中，成功地验证了反绎学习在此类复杂学习问题中的有效性。

本章的主要工作：

- W.-Z. Dai, Q.-L. Xu, Y. Yu, and Z.-H. Zhou. Tunneling neural perception and logic reasoning through abductive learning. **CoRR, abs/1802.01173**, 2018.

第六章　结束语

本论文中的内容主要涉及国家重点研发计划“大数据分析的基础理论和技术方法”（2018YFB1004300）以及英国皇家工程院牛顿基金合作研究计划“结合概率与逻辑表示的机器学习方法研究”（NRCP/1415/133）等项目。

在许多机器学习任务中，往往存在着一阶逻辑表达的领域知识。一方面，我们需要对这些领域知识加以利用，以提高机器学习性能；另一方面，我们还可以对领域知识进行精化，已获得具有良好可理解性的模型。在一些复杂情况下，机器学习与领域知识的推理和精化甚至相互影响和依赖。因此，如何将一阶逻辑表示的领域知识和机器学习相结合是一个重要的研究课题。多年以来，人工智能在知识推理和精化领域和机器学习领域均取得了长足进展，但关于如何将二者相结合的研究却相对较为薄弱。

本文对如何将一阶逻辑领域知识与机器学习相结合进行了探索。具体来说，针对该问题的三个方面，本文主要取得了以下创新成果：

在第2章中，本文提出了一种领域知识增广训练样本的机器学习方法。它可以将一阶逻辑表示的领域知识转化为机器学习可直接利用的特征或样例，以增加训练数据中所包含的信息，最终达到促进机器学习性能的目的。本文将此方法应用于统计关系学习任务中，并在机器学习的过程中引入了关于关系型路径的领域知识。实验结果表明，引入领域知识对样本进行增广后，机器学习的性能可以得到提升。

在第3章中，本文提出了一种领域知识辅助约束的机器学习方法。它可以将一阶逻辑表示的领域知识转化为机器学习优化过程中对假设空间的约束，以限制每轮优化得到的假设模型与领域知识保持一致。本文将此方法应用于图像特征学习任务中，并在学习过程中引入关于笔划的领域知识辅助特征提取。实验结果表明，该方法学得的特征在多个任务中均能取得更好的泛化性能，且能够学习出更符合领域知识描述的特征。

在第4章中，本文提出了一种机器学习驱动的知识精化方法。它将知识精化问题转化为一个机器学习问题，并结合常规机器学习方法与归纳逻辑程序实现一般问题中的知识精化。该方法首先使用反绎推理根据领域知识自动调用机

器学习模型从领域中提取基本逻辑事实，然后在这些事实的基础上通过机器学习方法进行知识精化。本文将该方法应用于图像分类任务，实验表明该方法可以从图像中学得具有良好可理解性，且能够直接进行重用的一阶逻辑知识。

在第5章中，本文提出了一种领域知识与机器学习互促结合的框架。它通过一致性优化将反绎逻辑推理与机器学习进行结合，其中机器学习能够从数据中自动学习提取伪标记对领域知识进行增广，而反绎逻辑推理则能够基于这些事实进行知识精化，一致性优化则与能够对判别伪标记正误的函数进行优化，以使得机器学习模型输出与领域知识保持最大一致。在复杂学习问题中实验结果表明，基于反绎学习的算法能够同时促进机器学习和知识精化。

基于本文在对领域知识与机器学习相结合的探索，将来的工作中还有许多有待进一步研究的问题：

引入一阶逻辑所表示的领域知识会对机器学习的效率产生影响。此类知识均拥有离散的结构，且往往十分复杂。当它们被引入机器学习时，一般需要通过离散搜索查找符合知识描述的逻辑事实以作为增广的样本或辅助约束。因此，我们很难在学习过程中引入大规模的领域知识库，或将领域知识应用在大规模的数据中。如何更有效率地对领域知识进行利用将是未来值得研究的一个方向。

而当使用领域知识对数据中的逻辑事实进行自动提取以增广领域知识时，目前的方法均只能提取知识中已定义的概念。然而在许多任务中，我们拥有的领域知识可能并不完全，对学习目标能够产生帮助的基本概念并不存在于事先拥有的领域知识中。谓词发明可能是一种有效的解决办法，但如何将其应用于非逻辑符号表示的数据中仍然有待探索。

本文研究的几个问题中均假设领域知识是正确的，即学习所得的结果一定不会与领域知识发生冲突。然而，在实际任务中我们无法保证这一点。并且，第3章的实验中的结果表明错误的领域知识甚至可能对机器学习的性能造成负面影响。所以如何在机器学习过程中自动发现并更正领域知识中的错误也是值得研究的问题之一。

参考文献

[1] S. B. Akers. Binary decision diagrams. *IEEE Transactions on computers*, 27(6): 509–516, 1978.

[2] D. Andrzejewski, X. Zhu, M. Craven, and B. Recht. A framework for incorporating general domain knowledge into Latent Dirichlet Allocation using first-order logic. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 1171–1177, Barcelona, Spain, 2011.

[3] F. Bacchus. Using first-order probability logic for the construction of bayesian networks. In *Proceedings of the 9th International Conference on Uncertainty in Artificial Intelligence*, pp. 219–219, Washihgton, DC, 1993.

[4] C. Baral and M. Gelfond. Logic programming and knowledge representation. *The Journal of Logic Programming*, 19-20:73 – 148, 1994. Special Issue: Ten Years of Logic Programming.

[5] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[6] E. Bellodi and F. Riguzzi. Learning the structure of probabilistic logic programs. In *Proceedings of the 21st International Conference on Inductive Logic Programming*, pp. 61–75, Dubrovnik, Croatia, 2012.

[7] E. Bellodi and F. Riguzzi. Experimentation of an expectation maximization algorithm for probabilistic logic programs. *Intelligenza Artificiale*, 6(1):3–18, 2012.

[8] E. Bellodi and F. Riguzzi. Structure learning of probabilistic logic programs by searching the clause space. *Theory and Practice of Logic Programming*, 15(2): 169 – 212, 2015.

[9] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning*, pp. 41–48, Montreal, Canada, 2009.

[10] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[11] A. Bharath and S. Madhvanath. Online handwriting recognition for indic scripts. In *Guide to OCR for Indic Scripts: Document Recognition and Retrieval*, pp. 209–234. 2010.

[12] M. Biba, S. Ferilli, and F. Esposito. Discriminative structure learning of Markov logic networks. In *Proceedings of the 18th International Conference on Inductive Logic Programming*, pp. 59–76, Prague, Czech Republic, 2008.

[13] C. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, NY, 2006.

[14] I. Bratko. *Prolog Programming for Artificial Intelligence*. Addison-Wesley Longman Publishing, Boston, MA, 1990.

[15] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[16] CACM Staff. What about statistical relational learning? *Communications of the ACM*, 58(12):8–9, 2015.

[17] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011.

[18] S.-C. Chou, X.-S. Gao, and J.-Z. Zhang. *Machine Proofs in Geometry: Automated Production of Readable Proofs for Geometry Theorems*. World Scientific, Singapore, 1994.

[19] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, 22 (12):3207–3220, 2010.

[20] K. L. Clark. *Negation as Failure*, pp. 293–322. Springer US, Boston, MA, 1978.

[21] K. L. Clark and F. McCabe. Prolog: a language for implementing expert systems. In *Machine Intelligence 10*, pp. 455–475. 1980.

[22] W. W. Cohen. PAC-learning a restricted class of logic programs. In *Proceedings of the 3rd International Workshop on Inductive Logic Programming*, pp. 41–72, London, UK, 1993.

[23] W. W. Cohen. Tensorlog: A differentiable deductive database. *CoRR*, abs/1605.06523, 2016.

[24] W. W. Cohen and C. D. Page. Polynomial learnability and Inductive Logic Programming: methods and results. *New Generation Computing*, 13:369–409, 1995.

[25] V. S. Costa, D. Page, M. Qazi, and J. Cussens. CLP(BN): Constraint logic programming for probabilistic knowledge. In *Proceedings of the 19th conference on Uncertainty in Artificial Intelligence*, pp. 517–524, Acapulco, Mexico, 2002.

[26] R. R. Curtin, J. R. Cline, N. P. Slagle, W. B. March, P. Ram, N. A. Mehta, and A. G. Gray. MLPACK: A scalable C++ machine learning library. *Journal of Machine Learning Research*, 14:801–805, 2013.

[27] J. Cussens. Parameter estimation in stochastic logic programs. *Machine Learning*, 44(3):245–271, 2001.

[28] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 13rd IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 886–893, San Diego, CA, 2005.

[29] E. Dantsin. Probabilistic logic programs and their semantics. In *Proceedings of the 1st Russian Conference on Logic Programming*, pp. 152–164, Moskow, Russia, 1992.

[30] A. Darwiche. A compiler for deterministic, decomposable negation normal form. In *18th National Conference on Artificial Intelligence*, pp. 627–634, Menlo Park, CA, 2002.

[31] L. De Raedt and M. Bruynooghe. CLINT: A multistrategy interactive concept-learner and theory revision system. In *Proceedings of the 1st International Workshop on Multistrategy Learning*, pp. 175–191, Harpers Ferry, WV, 1991.

[32] L. De Raedt and K. Kersting. Probabilistic inductive logic programming. In *Proceedings of the 15th International Conference on Algorithmic Learning Theory*, volume 3244, Padova, Italy, 2004.

[33] L. De Raedt, A. Kimmig, and H. Toivonen. Problog: A probabilistic prolog and its application in link discovery. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 2462–2467, Hyderabad, India, 2007.

[34] L. De Raedt, P. Frasconi, K. Kersting, and S. H. Muggleton. *Probabilistic Inductive Logic Programming*. Springer-Verlag, Berlin, 2008.

[35] L. De Raedt, K. Kersting, A. Kimmig, K. Revoredo, and H. Toivonen. Compressing probabilistic prolog programs. *Machine Learning*, 70(2):151–168, 2008.

[36] P. Domingos and D. Lowd. *Markov Logic: An Interface Layer for Artificial Intelligence*. Morgan & Claypool Publishers, 2009.

[37] I. Donadello, L. Serafini, and A. S. d'Avila Garcez. Logic tensor networks for semantic image interpretation. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 1596–1602, Melbourne, Australia, 2017.

[38] R. Drechsler and B. Becker. *Binary Decision Diagrams: Theory and Implementation*. Springer Science & Business Media, 2013.

[39] A. Dries, A. Kimmig, W. Meert, J. Renkens, G. Van den Broeck, J. Vlasselaer, and L. De Raedt. Problog2: Probabilistic logic programming. In *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pp. 312–315, Porto, Portugal, 2015.

[40] S. Džeroski and N. Lavrač. *Relational Data Mining*. Springer, Berlin, 2001.

[41] S. Džeroski, S. H. Muggleton, and S. Russell. PAC-learnability of determinate logic programs. In *Proceedings of the 5th ACM Workshop on Computational Learning Theory*, pp. 128–135, New York, NY, 1992.

[42] S. Džeroski, S. H. Muggleton, and S. Russell. Learnability of constrained logic programs. In *Proceedings of the 6th European Conference on Machine Learning*, pp. 342–347, London, UK, 1993.

[43] H. B. Enderton. *A Mathematical Introduction to Logic*. Elsevier, Amsterdam, Netherlands, 2001.

[44] R. Evans and E. Grefenstette. Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research*, 61:1–64, 2018.

[45] D. Fierens, G. Van Den Broek, J. Renkens, D. Shterionov, B. Gutmann, I. Thon, G. Janssens, and L. De Raedt. Inference and learning in probabilistic logic programs using weighted boolean formulas. *Theory and Practice of Logic Programming*, 15(3):358 – 401, 2015.

[46] J. Fiser, P. Berkes, G. Orbán, and M. Lengyel. Statistically optimal perception and learning: from behavior to neural representations. *Trends in Cognitive Sciences*, 14(3):119–130, 2010.

[47] P. Flach and N. Lachiche. 1BC: A first-order Bayesian classifier. In *Proceedings of the 9th Workshop on Inductive Logic Programming*, pp. 92–103, Bled, Slovenia, 1999.

[48] M. V. M. França, G. Zaverucha, and A. S. d'Avila Garcez. Fast relational learning using bottom clause propositionalization with artificial neural networks. *Machine Learning*, 94(1):81–104, 2014.

[49] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pp. 1300–1309, Stockholm, Sweden, 1999.

[50] J. Fürnkranz, D. Gamberger, and N. Lavrač. *Foundations of Rule Learning*. Cognitive Technologies. Springer, Berlin, 2012.

[51] A. S. d. Garcez, K. B. Broda, and D. M. Gabbay. *Neural-Symbolic Learning Systems: Foundations and Applications*. Springer-Verlag, London, 2012.

[52] H. Gelernter. Realization of a geometry theorem proving machine. In *Proceedings of IFIP Congress*, pp. 273–281, Paris, France, 1959.

[53] L. Getoor. *Learning Statistical Models from Relational Data*. PhD thesis, Stanford University, Stanford, CA, 2002.

[54] L. Getoor and B. Taskar, editors. *Introduction to Statistical Relational Learning*. MIT Press, Cambridge, MA, 2007.

[55] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, Cambridge, MA, 2016.

[56] G. Gottlob. Subsumption and implication. *Information Processing Letters*, 24 (2):109–111, 1987.

[57] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *CoRR*, abs/1410.5401, 2014.

[58] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwinska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, and J. Agapiou. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016.

[59] B. Gutmann, A. Kimmig, K. Kersting, and L. De Raedt. Parameter learning in probabilistic databases: A least squares approach. In *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pp. 473–488, Antwerp, Belgium, 2008.

[60] B. Gutmann, I. Thon, and L. De Raedt. Learning the parameters of probabilistic logic programs from interpretations. In *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pp. 581–596, Athens, Greece, 2011.

[61] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[62] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer New York Inc., New York, NY, 2001.

[63] D. R. Hofstadter. *Metamagical Themas: Questing for the Essence of Mind and Pattern*. Basic Books, Inc., New York, NY, 1985.

[64] S. Hölldobler and Y. Kalinke. Towards a new massively parallel computational model for logic programming. In *Working Notes of ECAI on Combining Symbolic and Connectionist processing*, pp. 1–14, Amsterdam, The Netherlands, 1994.

[65] M. Horsch and D. L. Poole. A dynamic approach to probabilistic inference using bayesian networks. In *Proceedings of the 6th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 155–161, Boston, MA, 1990.

[66] T. N. Huynh and R. J. Mooney. Discriminative structure and parameter learning for markov logic networks. In *Proceedings of the 25th International Conference of Machine Learning*, pp. 416–423, Helsinki, Finland, 2008.

[67] T. N. Huynh and R. J. Mooney. Max-margin weight learning for markov logic networks. In *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pp. 564–579, Bled, Slovenia, 2009.

[68] T. N. Huynh and R. J. Mooney. Online max-margin weight learning with markov logic networks. In *Proceedings of the 6th AAAI Workshop on Statistical Relational Artificial Intelligence*, pp. 32–37, Atlanta, GA, 2010.

[69] P. Idestam-Almquist. Learning missing clauses by inverse resolution. In *Proceedings of the 5th International Conference on Generation Computer Systems*, pp. 610–617, Tokyo, Japan, 1992.

[70] K. Inoue, K. Furukawa, I. Kobayashi, and H. Nabeshima. Discovering rules by meta-level abduction. In *Proceedings of the 9th International Conference on Inductive Logic Programming*, pp. 49–64, Leuven, Belgium, 2010.

[71] P. Jackson. *Introduction to Expert Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 3rd edition, 1998.

[72] R. Jia and P. Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2011–2021, Copenhagen, Denmark, 2017.

[73] A. C. Kakas, R. A. Kowalski, and F. Toni. Abductive logic programming. *Journal of Logic Computation*, 2(6):719–770, 1992.

[74] K. Kersting and L. De Raedt. Towards combining inductive logic programming with Bayesian networks. In *Proceedings of the 11th International Conference on Inductive Logic Programming*, pp. 118–131, Windsor Great Park, UK, 2001.

[75] T. Khot, S. Natarajan, K. Kersting, and J. Shavlik. Gradient-based boosting for statistical relational learning: the Markov logic network and missing data cases. *Machine Learning*, 100(1):75–100, 2015.

[76] J. Kietz. Some lower bounds on the computational complexity of inductive logic programming. In *Proceedings of the 6th European Conference on Machine Learning*, pp. 115–123, Vienna, Austria, 1993.

[77] A. Kimmig, L. Mihalkova, and L. Getoor. Lifted graphical models: A survey. *Machine Learning*, 99(1):1–45, 2015.

[78] R. D. King, K. E. Whelan, F. M. Jones, P. K. G. Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell, and S. G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247–252, 2004.

[79] R. D. King, J. Rowland, S. G. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, M. Markham, P. Pir, L. N. Soldatova, A. Aparkes, K. E. Whelan, and A. Clare. The automation of science. *Science*, 324(5923):85–89, 2009.

[80] S. Kok and P. Domingos. Learning the structure of markov logic networks. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 441–448, Bonn, Germany, 2005.

[81] S. Kok and P. Domingos. Learning markov logic network structure via hypergraph lifting. In *Proceedings of the 26th International Conference on Machine Learning*, pp. 505–512, Montreal, Canada, 2009.

[82] S. Kok and P. Domingos. Using structural motifs for learning markov logic networks. In *Proceedings of the 6th AAAI Workshop on Statistical Relational Artificial Intelligence*, pp. 46–51, Atlanta, GA, 2010.

[83] D. Koller and N. Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, Cambridge, MA, 2009.

[84] D. Koller and A. Pfeffer. Learning probabilities for noisy first-order rules. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, pp. 1316–1323, Nagoya, Japan, 1997.

[85] S. K. Kopparapu and V. L. Lajish. A framework for on-line devanagari handwritten character recognition. *CoRR*, abs/1410.6909, 2014.

[86] R. A. Kowalski. Predicate logic as programming language. In *Proceedings of IFIP Congress*, pp. 569–574, Stockholm, Sweden, 1974.

[87] R. A. Kowalski and D. Kuehner. Linear resolution with selection function. *Artificial Intelligence*, 2(3/4):227–260, 1971.

[88] S. Kramer. Predicate invention: A comprehensive view. Technical Report OFAI-TR-95-32, Austrian Research Institute for Artificial Intelligence, Vienna, Austria, 1995.

[89] S. Kramer, B. Pfahringer, and C. Helma. Stochastic propositionalisation of non-determinate background knowledge. In *Proceedings of the 8th International Conference on Inductive Logic Programming*, pp. 80–94, Madison, WI, 1998.

[90] S. Kramer, N. Lavrač, and P. Flach. Propositionalisation approaches to Relational Data Mining. In *Relational Data Mining*, pp. 262–291. Berlin, 2001.

[91] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pp. 1097–1105, 2012.

[92] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[93] N. Lavrač and S. Džeroski. *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, New York, NY, 1994.

[94] N. Lavrač, S. Džeroski, and M. Grobelnik. Learning non-recursive definitions of relations with LINUS. In *Proceedings of the 5th European Working Session on Learning*, Helsinki, Finland, 1991.

[95] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[96] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems 19*, pp. 801–808. 2007.

[97] C.-C. Liu, S. J. Lee, and S. S. Venkata. An expert system operational aid for restoration and loss reduction of distribution systems. *IEEE Transactions on Power Systems*, 3(2):619–626, 1988.

[98] J. W. Lloyd. *Logic for Learning*. Springer, Berlin, 2003.

[99] D. Lowd and P. Domingos. Efficient weight learning for markov logic networks. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp. 200–211, Warsaw, Poland, 2007.

[100] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[101] L. Magnani. *Abductive Cognition: The Epistemological and Eco-Cognitive Dimensions of Hypothetical Reasoning*. Springer-Verlag, Berlin, German, 1st edition, 2009.

[102] S. Mei, J. Zhu, and J. Zhu. Robust RegBayes: Selectively incorporating first-order logic domain knowledge into Bayesian models. In *Proceedings of the 31th International Conference on Machine Learning*, pp. 253–261, Beijing, China, 2014.

[103] D. Michie. Machine learning in the next five years. In *Proceedings of 3rd European Working Session on Learning*, pp. 107–122, Glasgow, UK, 1988.

[104] L. Mihalkova and R. J. Mooney. Bottom-up learning of markov logic network structure. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 625–632, Corvalis, OR, 2007.

[105] S. H. Muggleton. Duce, an oracle based approach to constructive induction. In *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pp. 287–292, Milan, Italy, 1987.

[106] S. H. Muggleton. Inverting the resolution principle. In *Machine Intellience 12*, pp. 93–104, 1991.

[107] S. H. Muggleton. Inductive Logic Programming. *New Generation Computing*, 8(4):295–318, 1991.

[108] S. H. Muggleton. Inverting implication. In *Proceedings of the 2nd Inductive Logic Programming Workshop*, pp. 19–39, Tokyo, Japan, 1992.

[109] S. H. Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13:245–286, 1995.

[110] S. H. Muggleton. Stochastic logic programs. *Advances in inductive logic programming*, 32:254–264, 1996.

[111] S. H. Muggleton. Learning stochastic logic programs. *Electronic Transactions on Artificial Intelligence*, 4(B):141–153, 2000.

[112] S. H. Muggleton and W. Buntine. Machine invention of first-order predicates by inverting resolution. In *Proceedings of the 5th International Conference on Machine Learning*, pp. 339–352, Ann Arbor, MI, 1988.

[113] S. H. Muggleton and C. Feng. Efficient induction of logic programs. In S. H. Muggleton, editor, *Inductive Logic Programming*, pp. 281–298. London, 1992.

[114] S. H. Muggleton, D. Lin, N. Pahlavi, and A. Tamaddoni-Nezhad. Meta-interpretive learning: Application to grammatical inference. *Machine Learning*, 94(1):25–49, 2014.

[115] S. Natarajan, T. Khot, K. Kersting, B. Gutmann, and J. Shavlik. Gradient-based boosting for statistical relational learning: The relational dependency network case. *Machine Learning*, 86(1):25–56, 2012.

[116] S. Natarajan, T. Khot, K. Kersting, B. Gutmann, and J. W. Shavlik. Gradient-based boosting for statistical relational learning: The relational dependency network case. *Machine Learning*, 86(1):25–56, 2012.

[117] J. Neville, D. Jensen, L. Friedland, and M. Hay. Learning relational probability trees. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 625–630, Washington, DC, 2003.

[118] A. Newell and H. A. Simon. Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, 19(3):113–126, 1976.

[119] A. Newell, J. C. Shaw, and H. A. Simon. Report on a general problem solving program. In *Proceedings of IFIP Congress*, volume 256, pp. 64, Paris, France, 1959.

[120] R. Ng and V. S. Subrahmanian. Probabilistic logic programming. *Information and Computation*, 101(2):150–201, 1992.

[121] S. Nienhuys-Cheng and R. De Wolf. *Foundations of Inductive Logic Programming*. Springer-Verlag, Berlin, 1997. LNAI 1228.

[122] N. J. Nilsson. Probabilistic logic. *Artificial intelligence*, 28(1):71–87, 1986.

[123] J. Nville and D. D. Jensen. Relational dependency networks. *Journal of Machine Learning Research*, 8:653–692, 2007.

[124] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[125] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[126] D. Page and A. Frisch. Generalization and learnability: A study of constrained atoms. In S. H. Muggleton, editor, *Inductive Logic Programming*. London, 1992.

[127] S. C. Peirce. Abduction and induction. In J. Buchler, editor, *Philosophical Writings of Peirce*. New York, NY, 1955.

[128] G. Plotkin. A note on inductive generalisation. In *Machine Intelligence 5*, pp. 153–163. 1970.

[129] G. D. Plotkin. A further note on inductive generalization. In *Machine Intelligence 6*. 1971.

[130] G. D. Plotkin. *Automatic Methods of Inductive Inference*. PhD thesis, Edinburgh University, 1971.

[131] D. Poole. Logic programming, abduction and probability. *New Generation Computing*, 11(3):377–400, 1993.

[132] D. Poole. The independent choice logic for modelling multiple agents under uncertainty. *Artificial Intelligence*, 94(1):7 – 56, 1997.

[133] D. Poole. First-order probabilistic inference. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pp. 985–991, Acapulco, Mexico, 2003.

[134] H. Poon and P. Domingos. Joint inference in information extraction. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pp. 913–918, Vancouver, Canada, 2007.

[135] R. Popplestone. An experiment in automatic induction. In B. Meltzer and D. Michie, editors, *Machine Intelligence 5*, pp. 203–215. Edinburgh, 1969.

[136] T. Pyszczynski and J. Greenberg. Toward an integration of cognitive and motivational perspectives on social inference: A biased hypothesis-testing model. *Advances in Experimental Social Psychology*, 20:297 – 340, 1987.

[137] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, CA, 1993.

[138] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 759–766, Corvalis, OR, 2007.

[139] B. L. Richards and R. J. Mooney. Learning relations by pathfinding. In *Proceedings of the 10th National Conference on Artificial Intelligence*, pp. 50–55, San Jose, CA, 1992.

[140] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

[141] J. A. Robinson. A machine-oriented logic based on the resolution principle. *Jounal of the ACM*, 12(1):23–41, 1965.

[142] T. Rocktäschel and S. Riedel. End-to-end differentiable proving. In *Advances in Neural Information Processing Systems 30*, pp. 3791–3803. 2017.

[143] T. Rocktäschel, S. Singh, and S. Riedel. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1119–1129, Denver, CO, 2015.

[144] C. Rouveirol and J. Puget. A simple and general solution for inverting resolution. In *Proceedings of the 4th European Working Session on Learning*, pp. 201–210, London, UK, 1989.

[145] S. J. Russell. Unifying logic and probability. *Communications of the ACM*, 58 (7):88–97, 2015.

[146] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, London, UK, 3rd edition, 2016.

[147] T. Sato. A statistical learning method for logic programs with distribution semantics. In *In Proceedings of the 12th International conference on Logic Programming*, pp. 715–729, Tokyo, Japan, 1995.

[148] T. Sato. Generative modeling with failure in prism. In *International Joint Conference on Artificial Intelligence*, pp. 847–852, Edinburgh, UK, 2005.

[149] T. Sato and Y. Kameya. Prism: a language for symbolic-statistical modeling. In *In Proceedings of the 15th International Joint Conference on Artificial Intelligence*, pp. 1330–1335, Nagoya, Japan, 1997.

[150] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[151] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *Proceedings of the 2nd International Conference on Learning Representations*, Banff, Canada, 2014.

[152] E. Y. Shapiro. Logic programs with uncertainties: A tool for implementing rule-based systems. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, volume 83, pp. 529–532, Karlsruhe, West Germany, 1983.

[153] H. A. Simon and A. Newell. Human problem solving: The state of the theory in 1970. *American Psychologist*, 26(2):145, 1971.

[154] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, 2015.

[155] R. Socher, D. Chen, C. D. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems 26*, pp. 926–934. 2013.

[156] L. Sterling and R. D. Beer. Metainterpreters for expert system construction. *The Journal of Logic Programming*, 6(1):163 – 178, 1989.

[157] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pp. 485–492, Edmonton, Canada, 2002.

[158] M. Thoma. The HASYv2 dataset. *CoRR*, abs/1701.08380, 2017. URL http://arxiv.org/abs/1701.08380.

[159] C. Tiercelin. Abduction and the semiotics of perception. *Semiotica*, 2005(153): 389–412, 2005.

[160] G. G. Towell and J. W. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70(1):119 – 165, 1994.

[161] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11): 1134–1142, 1984.

[162] J. Van Haaren, G. Van den Broeck, W. Meert, and J. Davis. Lifted generative learning of markov logic networks. *Machine Learning*, 103(1):27–55, 2016.

[163] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. 2017.

[164] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for MATLAB. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, Brisbane, Australia, 2015.

[165] J. Vennekens, S. Verbaeten, and M. Bruynooghe. Logic programs with annotated disjunctions. In *Proceedings of the International Conference on Logic Programming*, pp. 431–445, Saint Malo, France, 2004.

[166] J. Vennekens, M. Denecker, and M. Bruynooghe. CP-logic: A language of causal probabilistic events and its relation to logic programming. *Theory and Practice of Logic Programming*, 9(3):245 – 308, 2009.

[167] S. Vere. Induction of concepts in the predicate calculus. In *Proceedings of the 4th International Joint Conference on Artificial Intelligence*, pp. 282–287, Tbilisi, USSR, 1975.

[168] W. Y. Wang, K. Mazaitis, and W. W. Cohen. Programming with personalized pagerank: a locally groundable first-order probabilistic logic. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pp. 2129–2138, San Francisco, CA, 2013.

[169] M. P. Wellman, J. S. Breese, and R. P. Goldman. From knowledge bases to decision models. *Knowledge Engineering Review*, 7(1):35–53, 1992.

[170] J. Weston, S. Chopra, and A. Bordes. Memory networks. *CoRR*, abs/1410.3916, 2014.

[171] J. Wielemaker, T. Schrijvers, M. Triska, and T. Lager. SWI-Prolog. *Theory and Practice of Logic Programming*, 12(1-2):67–96, 2012.

[172] J. Wnek and R. Michalski. Hypothesis-driven constructive induction in AQ17-HCI: A method and experiments. *Machine Learning*, 14(2):139–168, 1994.

[173] F. Yang, Z. Yang, and W. W. Cohen. Differentiable learning of logical rules for knowledge base reasoning. In *Advances in Neural Information Processing Systems 30*, pp. 2319–2328. 2017.

[174] K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. In *Advances in Neural Information Processing Systems 22*, pp. 2223–2231. 2009.

[175] Y. Yu, H. Qian, and Y.-Q. Hu. Derivative-free optimization via classification. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 2286–2292, Phoenix, AZ, 2016.

[176] H. Zhang, J. E. Fritts, and S. A. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, 2008.

[177] Z.-H. Zhou. Learnware: on the future of machine learning. *Frontiers of Computer Science*, 10(4):589–590, 2016.

[178] Z.-H. Zhou. Abductive learning: Towards bridging machine learning and logical reasoning. *Science China Information Sciences*, 2019. in press.

[179] 吴文俊. 数学机械化. 科学出版社, 北京, 中国, 2003.

[180] 周志华. 机器学习. 清华大学出版社, 北京, 中国, 2016.

致　　谢

2009 年前第一次来到南京的光景还历历在目。当时，我惴惴不安地来到南京大学，并第一次踏进了蒙民伟楼 LAMDA 研究所的大门。经过一年的努力，我终于踏踏实实地在大楼里拥有了自己的座位。如今，八年多的时间如白驹过隙，我的博士研究生生涯即将结束。回望这些年的种种，我收获到不少成功和雀跃，也品尝了许多失败与懊悔。这段岁月里，我见到了自己，见到了天地。这一切给我带来的成长，离不开每一位师长、朋友和家人对我的无私帮助。

感谢我的恩师周志华教授。八年来，周老师对我学业的支持，生活的关心，是我今天能顺利完成论文的前提。八年来，老师对我为人处世的教导，独立思维的培养，则是我未来一生能在社会中安身立命的凭借。在周老师面前，我逐渐看清并认识了自己。老师的睿智和博学则无时不让我感到折服，而老师的谦逊和正直则更是让我景仰。在科研上，周老师总是对我悉心指导。从研究的选题、实验的设计、论文的写作到报告的准备，老师都充满耐心地帮助我调整和修改。当科研遇到瓶颈时，周老师会鼓励我走出实验室去工业界实习，并给予我机会访问世界上最一流的学者并建立合作。此外，周老师也经常给予我参与事务工作的机会，让我获得了大量实际经验与并锻炼了工作技能。在为人处世方面，老师总能够一针见血地指出我的缺陷。打小以来，我性格中的坏毛病不仅令我在生活上给身边的人们带来困扰，更影响了我自己的科研和学业。为了帮助我改正，周老师对我总是一次又一次地宽容，并不厌其烦地倾听我的苦恼，反复而耐心地对我进行教育和劝导。在 LAMDA 学习生活的八年，周老师为培养我而付出的心血，只会比我意识到的更多。

感谢姜远老师。她一直以来对我生活上的关心，让我在远离家乡的南京，时常感受到如图家中一般的温暖。感谢俞扬老师在我刚进组时对我的指引，他对人工智能的热情和科研的认真将永远是我在学术道路上前行的榜样。感谢黎铭老师、詹德川老师、王魏老师、高魏老师和李宇峰老师，感谢他们在科研和工作上对我进行的无私帮助，并在我遇见挫折的时候的安慰和鼓励。

感谢我在访问伦敦帝国理工学院期间指导我的 Stphen Muggleton 教授。作为归纳逻辑程序设计领域的顶尖专家，Muggleton 教授在科研和学习上给予了

我大量帮助，让我这样一个曾完全不了解归纳逻辑程序设计的学生逐渐变得有能力参与此领域的工作。并且，Muggleton 教授在生活中也是一位和蔼可亲的老人，与他交流的过程让我第一次感受到英国的绅士风度，他展现的智慧与博学也由衷让我感到钦佩。感谢萨里大学的 Alireza Tamaddoni-Nezhad 老师和牛津大学的 Andrew Cropper 博士，他们在我访问帝国理工学院期间带我熟悉校园、出门游玩，并热心地帮助我适应国外的生活。

感谢所有 LAMDA 的师兄师姐和师弟师妹们，他们对我的帮助和关心让我的这八年博士生生涯得以顺利完成，他们的可爱和有趣也让这八年生活充满了欢笑和色彩。感谢李楠师兄、黄圣君师兄、钱超师兄、徐淼师姐、李绍园、张腾、朱越和丁尧相在科研、工作和生活中对我的种种帮助，与他们成为朋友是我博士生生涯中最宝贵的收获之一。特别感谢胡毅奇、杨杨、王璐、沈芷玉、侯博建、赵鹏、苗园、范林鹏、王少博、宋歌、张振宇和史舒婷等师弟师妹们在科研之外工作中对我的帮助，他们的热情、积极和靠谱，为我分担了许多我独力难以完成的任务。

感谢我的妻子刘雅静女士。她的默默支持和无私奉献是支撑我完成博士学习最坚定的力量，也是我未来生活中继续前行时最强大的动力。

我要由衷的感谢我辛劳的母亲和父亲，感谢他们在这二十几年里对我的养育之恩。正是由于他们对我生活上的关心和照顾、学业上的支持，才使我能安心学习和研究。在他们面前，我从未感受到任何压力，只有对我无限的包容。

最后，我要感谢我即将出生的孩子。你将是我的生命之光，是我新生活中奋斗的动力。感谢你让我感受到即将成为人父的欣喜以及沉甸甸的幸福责任。

依靠着大家的保护和帮助，我磕磕绊绊地行了这么多的路，总算有所成长。临别虽然不舍，但时间照例不许我停下脚步。我将把从你们那里得到的爱，化作责任和力量，担在肩上传下去。

附录 A　攻读博士学位期间的学术成果和获奖情况

一、攻读博士学位期间撰写的主要论文

已发表论文

1. W.-Z. Dai and Z.-H. Zhou. Combining logic abduction and statistical induction: Discovering written primitives with human knowledge. **In: Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'17)**, San Francisco, CA, 2017, pp.4392-4398. （*CCF-A* 类会议）
2. W.-Z. Dai and Z.-H. Zhou. Statistical unfolded logic learning. **In: Proceedings of the 7th Asian Conference on Machine Learning (ACML'15)**, Hong Kong, China, 2015, JMLR: W&CP 45, pp. 349-361. （机器学习领域重要国际会议）
3. W.-Z. Dai, S. H. Muggleton, J. Wen, A. Tamaddoni-Nezhad and Z.-H. Zhou. Logical vision: One-Shot Meta-Interpretive Learning on Real Images. **In Proceedings of the 27th International Conference on Inductive Logic Programming (ILP' 17)**, Orleans, France, 2017, pp.46-62. *(*归纳逻辑程序设计领域旗舰会议*)*
4. W.-Z. Dai, S. H. Muggleton, and Z.-H. Zhou. Logical vision: Meta-interpretive learning for simple geometrical concepts. **In Proceedings of the 25th International Conference on Inductive Logic Programming (ILP' 15)**. Kyoto, Japan, 2015, pp.1-16. *(*归纳逻辑程序设计领域旗舰会议*)*
5. 戴望州，周志华. 归纳逻辑程序设计综述. **计算机研究与发展**. 56(1): 138-154, 2019. *(*计算机类核心期刊*)*
6. S. H. Muggleton, W.-Z. Dai, C. Sammut, A. Tamaddoni-Nezhad, J. Wen, and Z.-H. Zhou. Meta-interpretive learning from noisy images. **Machine Learning**. 107(7): 749-766, 2018. *(*机器学习领域重要国际期刊*)*

其他公开论文

7. W.-Z. Dai, Q.-L. Xu, Y. Yu, and Z.-H. Zhou. Tunneling Neural Perception and Logic Reasoning through Abductive Learning. **CoRR, abs/1802.01173**, 2018.

二、攻读博士学位期间获得的主要奖励

1. IBM 博士生英才计划奖（IBM PhD Fellowship Award），2016 年
2. 谷歌优秀奖学金（Google Excellence Fellowship），2012 年
3. 南京大学优秀研究生，2012 年
4. 南京大学优秀博士生提升计划，2015 年
5. AAAI 学生旅行奖，2017 年
6. ILP 学生旅行奖，2015 年

三、攻读博士学位期间参加的主要科研项目

1. 国家重点研发计划“大数据分析的基础理论和技术方法”（2018YFB1004300）
2. 英国皇家工程院牛顿基金合作研究计划“结合概率与逻辑表示的机器学习方法研究”（NRCP/1415/133）