Author：Liu Jian

Time：2020-07-02

机器学习13-规则学习

规则学习 (rule learning) 是符号主义学习 (symbolism learning) 的主要代表。规则学习里的规则指的是逻辑规则。相比黑箱学习，规则学习有更好的可解释性。数理逻辑具有极强的表达能力，可以自然地在学习过程中引入领域知识。此外，逻辑规则的抽象描述能力在处理一些高度复杂的 AI 任务时具有显著的优势，比如问答系统中有可能遇到非常多、甚至无穷种可能的答案，此时若能基于逻辑规则进行抽象表述或者推理则将带来极大的便利。规则学习的目标就是从训练集中学得规则集合。

概念：

1. 符合某个规则的样本称为被该规则覆盖 (cover)。
2. 当同一个示例被判别结果不同的多条规则覆盖时，称发生了冲突 (conflict)，解决冲突的方法称为冲突消解 (conflict resolution)，常见策略有：投票法 (多数胜出)、排序法 (使用优先级高的规则的结果)、元规则法 (根据领域知识事先设定一些 meta-rule，即关于规则的规则，比如，“发生冲突时使用长度最小的规则”)等。
3. 默认规则 (default rule)：用来处理规则集合未覆盖的样本。

规则学习的规则可分为：命题规则 (propositional rule)、一阶规则 (first-order rule，也称关系型规则 relational rule) 和归纳逻辑程序设计 (inductive logic programming, ILP)。**命题规则是一阶规则的特例，而 ILP 则在一阶规则学习中引入了函数和逻辑表达式的嵌套，可以看到，表达能力：命题规则 < 一阶规则 < ILP。**

> 命题规则举例：

$$\text{好瓜} \leftarrow (\text{脐部} = \text{凹陷}) \wedge (\text{根蒂} = \text{蜷缩})$$

> 一阶规则举例：

$$(\forall X, \forall Y)(\text{更好}(X, Y) \leftarrow \text{脐部更凹}(X, Y) \wedge \text{根蒂更蜷}(X, Y))$$

下面，我们将简要地介绍命题规则、一阶规则、ILP 的学习策略。

1 规则学习的基本策略 - 以命题规则为例

本章包括**序贯覆盖**和**剪枝优化**两个部分。序贯覆盖给出了规则的生成策略，类似于决策树，剪枝优化通过对规则进行剪枝，以降低过拟合的风险。

1.1 序贯覆盖

序贯覆盖 (sequential covering)，即逐条归纳，采用的是分治策略 (separate-and-conquer)：在训练集上每学到一条规则，就将就将该规则覆盖的训练样例去掉，然后以剩下的训练样例组成训练集重复上述过程。

规则的提出会面临组合爆炸的问题，和以往一样，我们采用贪心的策略来产生单个规则：

1. 自顶向下：从比较一般的规则开始，逐渐添加新文字以缩小规则覆盖范围，直到满足预定条件为止，是规则逐渐特化 (specialization) 的过程。
2. 自底向上：从比较特殊的规则开始，逐渐删除文字以扩大规则覆盖范围，直到满足条件为止，是规则逐渐泛化 (generalization) 的过程。

自顶向下的策略更容易产生泛化性能较好的规则，对噪声的鲁棒性比自底向上强得多；自底向上策略的优点是更适用于训练样本较少的情况。因此，在命题规则学习中，通常适用自顶向下的策略，而在一阶规则学习这类假设空间非常复杂的任务上，自底向上的策略使用较多。

可以看到，一条规则就类似于决策树中的叶子结点，给出了落在这条规则 (结点) 里面的输入的标记，区别在于：

1. 叶子结点覆盖的训练样例可能含有多个标记 (当然用于预测时只会输出一个最可能的标记)，而根据上述策略生成规则时，规则覆盖的训练样本的标记相同；
2. 两条规则所描述输入 $\boldsymbol{x}$ 的范围中，某些属性的取值范围可能存在相交的部分，而各叶子结点对应输入 $\boldsymbol{x}$ 的范围是相互不相交的。

1.2 剪枝优化

类似于决策树，我们通过剪枝来减小过拟合的风险，同样地，存在预剪枝和后剪枝两种策略。

剪枝的对象：

1. 可以是单个规则中的文字；
2. 也可以是规则集中的某个规则；

而是否进行剪枝，则常基于某种性能度量指标来评估增/删逻辑文字前后的规则性能，或增/删规则前后的规则集性能。

剪枝还可借助统计显著性检验来进行。比如 CN2 算法在预剪枝时，假设用规则集进行预测必须显著优于直接基于训练样例集后验概率分布进行预测，为此，CN2 使用似然率统计量 (likelihood ratio statistics, LRS) 进行判断，而 LRS 实际上也是一种信息量指标。

后剪枝常用的策略有是错剪枝 (reduced error pruning, REP)。REP 的过程是：将数据集划分为训练集和验证集，在训练集上学得规则集 $\mathcal{R}$ 后进行多轮剪枝：每一轮穷举所有可能的剪枝操作，然后用验证集进行评估，保留最好的规则集进入下一轮剪枝，直到无法通过剪枝提高验证集上的性能为止。

REP 的复杂度较高，而 IREP (incremental REP) 则降低了复杂度：在生成下一条规则前，先将数据集划分为训练集和验证集，在训练集上生成一条规则 r ，并立即在验证集上对其进行 REP 剪枝，得到规则 r' ；将 r' 覆盖的样例去除，在更新后的数据集上重复上述过程。REP 和 IREP 的对比：

1. REP 对规则集进行剪枝，而 IREP 仅对单条规则进行剪枝，所以复杂度降低了；
2. REP 为后剪枝，而 IREP 为预剪枝。

RIPPER 算法则是在 IREP* (IREP 的改进，区别在于所用规则性能度量的指标不同) 所得规则集 $\mathcal{R}$ 上作如下的进一步优化：对 $\mathcal{R}$ 中的每条规则 r_i ，RIPPER 为它产生两个变体：

- 替换规则 (replacement rule) r_i' ：基于 r_i 覆盖的样例，用 IREP* 重新生成一条规则 r_i' ；
- 修订规则 (revised rule) r_i'' ：对 r_i 增加文字进行特化，然后再用 IREP* 剪枝生成一条规则 r_i'' ；

接下来，把 r_i' 和 r_i'' 分别与 $\mathcal{R}$ 中除 r_i 之外的规则放在一起，组成规则集 $\mathcal{R}'$ 和 $\mathcal{R}''$ ，将它们与 $\mathcal{R}$ 一起进行比较，选择最优的规则集保留下来。

2 一阶规则的学习

命题规则学习难以处理对象之间的关系，而关系信息在很多任务中非常重要；此外，相对于命题规则学习，一阶规则学习能容易地引入领域知识。

著名的一阶规则学习算法有 FOIL (first-order inductive learner)：

1. 整体上，它遵循**序贯覆盖**框架且采用**自顶向下**的规则归纳策略。
2. 在生成单条规则时 (即局部上)，FOIL 使用 FOIL 增益 (FOIL gain) 来选择文字 (类似于构建决策树时使用信息增益进行属性选择)，直到规则长度增加合适长度为止 比如能够在处理训练样本时不出错时为止。FOIL 增益：

$$\mathrm{F\_Gain} = \hat{n}_+ \left(\log_2 \frac{\hat{n}_+}{\hat{n}_+ + \hat{n}_-} - \log_2 \frac{n_+}{n_+ + n_-} \right)$$

其中，n_+ 和 n_- 分别表示增加某个文字前，该条规则在训练集中正确处理与错误处理的样本个数；$\hat{n}_+$ 和 $\hat{n}_-$ 分别表示增加某个文字后，该条规则在训练集中正确处理与错误处理的样本个数；显然，添加文字前后，训练集中可被该条规则处理的样本个数并不一定相同，即 $\hat{n}_+ + \hat{n}_- \neq n_+ + n_-$ 。

3. 最后，FOIL 使用后剪枝对规则集进行优化。

若允许将目标谓词作为候选文字加入规则体，则 FOIL 能学出递归规则；若允许将否定形式的文字作为候选，则往往能得到更简洁的规则集。

FOIL 自顶向下的规则生成过程不能支持函数和逻辑表达式嵌套，因此相比 ILP 表达能力仍有不足，但是它把命题规则学习过程通过变量替换等操作直接转化为一阶规则学习，因此比一般归纳逻辑程序设计技术更高效。

3 归纳逻辑程序设计

归纳逻辑程序设计 (ILP) 在一阶规则学习中引入了函数和逻辑表达式嵌套。由于 ILP 学得的规则几乎能直接被 PROLOG 等逻辑程序解释器调用，而 PROLOG 在专家系统中常被使用，因此 ILP 成为连接机器学习与知识工程的重要桥梁。IPL 在生物数据挖掘和自然语言处理等任务中取得了一些成功，但其表示能力太强，直接导致学习过程面临的假设空间太大，复杂度极高，因此问题规模稍大就难以有效进行学习。近年来，随着机器学习技术进入更多应用领域，在富含结构信息和领域知识的任务中，逻辑表达的重要性逐渐凸显，因此出现了一些将规则学习与统计学习相结合的努力，这成为机器学习发展的一大趋势。

归纳逻辑程序设计采用自底向上的规则生成策略，直接将一个或多个具体事实 (grounded fact) 作为初始规则，再对规则逐步进行泛化以增加其对样例的覆盖率；通过泛化得到单条规则后，我们将其加入规则集，最后进行剪枝等进一步优化。下面介绍两种泛化方法：最小一般泛化和逆归结。

3.1 最小一般泛化

泛化操作可以是将规则中的常量替换为逻辑变量，也可以是删除规则体中的某个文字，最小一般泛化 (least general generalization, LGG) 就是这样的一种泛化方法，具体细节可见西瓜书，不再赘述。此外，还有 RLGG (relative least general generalization)。

3.2 逆归结

逆归结涉及到更深一点的数理逻辑知识，离散数学中学过但很多都搞忘了，这使得逆归结是西瓜书上规则学习这章中唯一没有读懂的内容。不过不要紧，只要知道其作用是什么就可以了，具体细节可以需要的时候再补。

总的来说，逆归结也是一种规则的泛化方法，其一大特点是能自动发明新谓词，这些新谓词可能对应于样例属性和背景知识中不存在的新知识，对知识发现与精化有重要意义，但自动发明的新谓词究竟对应于什么语义，则只能通过使用者对任务领域的进一步理解才能明确。