目录

第十五章　规则学习

0.1　基本概念

从形式语言表达能力而言，规则可分为两类：**命题规则** (ropositional rule) 和**一阶规则** (first-order rule)。命题规则是由"原子命题"(propositional atom) 和逻辑连接词与、或、非和蕴含构成的简单陈述句。一阶规则 (first-order rule) 不像命题规则只处理简单的陈述命题，一阶逻辑还额外包含了断言和量化 (谓词和量词)，用来描述事物的属性和关系。一阶规则能表达复杂的关系，也被称为关系型规则 (relational rule)。

生成规则的方法也分为两种:

- 直接生成法 (Direct Method)：直接从训练集中归纳出规则
- 间接生产法 (Indirect Method)：从决策树转换而来

0.2　序列覆盖

序列覆盖 (Sequencial Covering) 即逐条归纳，属于直接生成法。在训练集上每学习到一条规则，就将该规则覆盖的训练样例去除，然后以剩下的样例组成训练集重复上述过程，由于每次只处理一部分数据，因此也被称为分治 (separate-and-conquer) 的策略。

0.2.1　规则产生的策略

- 自顶向下（top-down）：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，直到满足预定的条件为止。亦称"生成－测试"（generate-then-test）法，是规则逐渐"特化"（specialization）的过程。(类似决策树)
- 自底向上（bottom-top）：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，亦称数据驱动 (data-driven) 方法，是规则逐渐泛化 (generalazation) 的过程。

0.2.2　比较

- 第一种策略更容易产生泛化性能更好的规则，第二种策略更适合于训练样本较少的情况。
- 第一种策略对噪声的鲁棒性比后者强。命题规则学习中通常使用第一种策略.
- 第二种策略在一阶规则学习这类假设空间非常复杂的任务上使用较多。

0.3　剪枝优化

原因: 序列覆盖是一个贪心搜索的过程, 需要机制来**缓解过拟合**的风险. 决策树一样分为: 预剪枝和后剪枝.

常用算法:

1. PRISM 算法：第一个提出序列覆盖的算法, 基于准确率和覆盖率进行判断；
2. CN2 算法：第一个考虑过拟合的算法, 预剪枝, 基于 AQ 算法并结合 ID3 算法处理噪音数据, 用熵对结果进行判断, 熵越小则质量越高；
3. FOIL 算法：基于 CN2 算法的改进, 基于 FOIL 增益 (FOIL gain) 进行质量判断, 适用于一阶规则学习，其中 FOIL 增益为：

$$\text{F\_Gain} = \hat{m}_+ \times (\log_2 \frac{\hat{m}_+}{\hat{m}_+ + \hat{m}_-} - \log_2 \frac{m_+}{m_+ + m_-})$$

其中 $\hat{m}_+, \hat{m}_-$ 分别是增加候选文字后新规则所覆盖的正、反例数；m_+, m_- 为原规则覆盖的正、反例数。

4. REP 算法 (Reduce Error Pruning)：$O(m^4)$, 后剪枝, 用准确率作为剪枝指标, 先生成 Rule 集合, 再剪枝，具体是：将样例集划分为训练集和验证集，从训练集上学得规则集 R 后进行多轮剪枝，在每一轮穷举所有可能的剪枝操作，然后用验证集对剪枝产生的所有候选规则集进行比较，保留最好的规则集进行下一轮剪枝 (也可以理解为如果错误增加了, 就剪掉, 否则保留)。时间复杂度是 $O(m^4)$，m 为训练样例数目。**REP 是针对规则集进行剪枝。**
5. IREP 算法 (Incremental REP)：$O(m \log^2 m)$, 后剪枝, 相比 REP 高效, 每生成一个规则就进行剪枝验证：在生成每条规则之前，先将当前样例集划分为训练集和验证集，在训练集上生成一条规则 r，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，在更新后的样例集上重复上述过程。**IREP 仅对单条规则进行剪枝，因此更高效。**
6. REIPPER 算法：基于 IREP* 代替 IREP 的准确率, 防止局部最优, 泛化性能好, 学习速度比决策树更快。

0.4 间接方法

从决策树生成。