计算机科学2000Vol.27№.3

知识发现方法的比较研究

The Comparison of Methods Used in KDD

陆 伟 吴朝晖

（浙江大学人工智能研究所 杭州 310027）

Abstract Knowledge Discovery in Database(KDD) is a rapid emerging research field. This paper gives an analysis and comparison for methods used in Knowledge Discovery in Database. Our aim is to discuss how to utilize those methods properly in order to construct an efficient KDD system.

Keywords KDD, Model recognition, Machine learning, Rough set

1 引言

从已有信息中发现模式或规律是信息处理技术的本质所在。长期以来，为了实现这个目标人们从不同领域、不同角度提出各种方法。典型的如：从数学角度提出的数理统计方法、从模拟生物神经结构角度提出的神经网络技术、从知识角度提出的机器学习方法等，与上述方法不同的是，KDD 并不是研究某种具体的方法，而是根据用户的需要和领域的特点，利用已有的技术形成一个完整的系统，在有限的计算资源下从大型数据库中自动地发现知识。因此我们认为，KDD 着重于系统的实用性，其主要目的是使上述方法适用于大型数据库以及根据领域特性适当地利用它们。

本文主要从方法论的角度对 KDD 作一个初步的介绍。通过对 KDD 中各常用方法的介绍和比较，来讨论当前 KDD 各方法在不同应用背景下的优缺点。

2 KDD 概述及比较研究框架

KDD 的定义在不同应用领域有多种不同的解释，目前，比较为大家所接受的定义是由 Fayyad 等[1]给出的定义：

KDD 是从目标数据集合中识别出有效的、新颖的、潜在有用的，以及最终可理解的模式的非平凡过程。

整个 KDD 的过程可大致分为：数据准备、数据挖掘、知识评估和表现。数据准备阶段主要包括数据选择、数据清理和数据预处理。本质上，数据准备阶段也是一种粗粒度的模式发现过程（如预测空数据、概念层次变化和维数简约等），因此我们将要讨论的各种方法也适用于该阶段。数据挖掘阶段是整个 KDD 过程的核心，首先要确定需要从数据中发现什么类型的知识，数据挖掘任务可以是总结描述、分类、聚类、关联规则发现或序列模式发现等。确定了开采任务后，就要决定使用什么样的开采方法。由于领域对挖掘任务的约束条件千差万别，同时作为挖掘算法一部分的目标数据和领域知识本身存在着多种异质的表达方式，因此需要根据实际的挖掘任务和领域特点，来选择合适的挖掘算法。目前流行的 KDD 方法来源于多个领域，典型的如数理统计、机器学习、模式识别、神经网络、数据库技术等。数据挖掘往往可以发现相当数量的模式，因此一个完整的 KDD 系统应该能对发现的模式进行评估，挑选出其中能够成为知识的模式。本文对 KDD 各方法的比较研究框架是基于以下九个标准：

◇ 描述模型的能力：该方法是否能够从数据中挖掘出复杂的模型；

◇ 可伸缩性：该方法对目标数据集合的大小的敏感度，即是否适合于大型数据库；

◇ 精确性：该方法挖掘出的模型是否精确；

◇ 鲁棒性：该方法对非法输入、错误数据以及环境因素的适应能力；

◇ 抗噪音能力：如果目标数据中存在数据丢失、失真等情况时，该方法是否能够自动恢复正确的值或仅仅将噪音过滤；

◇ 知识的可理解性：该方法发现的知识是否能够为人所理解，是否能够作为先验知识被再利用；

◇ 是否需要主观知识：该方法在挖掘过程中，是否依赖于外部专家的主观知识；

◇ 开放性：该方法是否能够结合领域知识来高效地发现知识；

◇ 适用的数据类型：该方法是否只适用于数值类型的数据或是符号类型的数据或者皆可。

下面，将在该框架下对各类方法作简单的介绍。

3 统计模式识别方法

统计模式识别方法指运用数理统计的方法，在样本空间中抽取出模式。模式可以是对样本的分类、聚类、回归估计和分布密度估计。下面介绍统计模式识别中的几种方法：

1）几何分类　利用样本在样本空间的几何分布状况，找到类与类之间的分割曲面，在此基础上进行分类或聚类。典型的算法如：距离分类法、感知器、LMSE梯度法和势函数法。该类方法的关键在于：在搜索空间中找到分割曲面，同时保证曲面与类之间的距离足够大，避免出现过分偶合(Over-fit)。

几何分类器的优点在于不依赖条件概率分布密度知识，但它所能处理的只是确定可分的模式，当样本集聚的空间发生重叠现象时，寻找分类曲面的迭代过程将加长或产生振荡。

2）概率分类法　其基本思想是利用贝叶斯法则从样本数据中获得各个模式的后验概率，从后验概率出发，运用风险函数，求出具有最小风险的贝叶斯模式分类器。所谓风险函数是指将属于 y_i 类样品 X 决策为 y_i 类(记为 a_i)风险，记为 $\lambda(a_i|y_i)$。因此将样本 X 决策为 y_i 类的总风险为：

$$R(a_i|X)=\sum_{y_i}\lambda(a_i|y_i)P(y_i|X)$$

分类器选择决策风险 $R(a_i|X)$ 为最小的类作为样本 X 的分类。每个类的先验概率 $P(y_i)$ 代表系统对任务的先验知识；条件概率 $P(X|y_i)$ 的分布形式可以根据先验知识来估计而函数的参数向量可以从训练数据中估计出(估计的方法可以是最大似然估计、贝叶斯估计等方法)。

概率分类器的优点是训练样本不必几何可分，对噪音和畸变不敏感等；但它需要用户提供先验概率和条件概率分布形式，即使采用密度函数的参数估计仍然较粗糙，因此分类器的性能受主观知识的影响较大[2]。

3）动态聚类分析　聚类分析是在训练集中无教师信号情况下根据样本数据自身的分布特点来进行分类。动态聚类是聚类分析中常用的方法之一。其基本过程是：

- 随机选择若干样本为聚类中心；
- 按某种聚类准则，如最小距离准则，使其余样品向各中心聚集，得到一种聚类模式；
- 评价该聚类的优劣，若满意则结束算法，否则按照一定的策略重新选择样本作为聚类中心，重复上述步骤。其中选择聚类中心的策略可以是选择已有聚类的几何中心或选择离中心最远的边界点。

比较著名的动态聚类算法有 K 均值算法和 ISODATA 算法。

4）贝叶斯网　是一种用于描述对象之间的概率关系的有向非循环图模型，在数据挖掘领域中，主要用于发现对象间的因果关系[3]。贝叶斯网中的每个结点表示变量或状态，两点之间的有向弧表示结点之间的依赖关系。网络的结构实际上描述了变量之间的条件独立性(两点之间若不存在弧，表明两个变量条件独立)，而每个变量都有相应的局部概率分布。以此为基础，可以计算出整个变量集合或变量子集的联合分布密度。一个典型的贝叶斯网如图1所示。

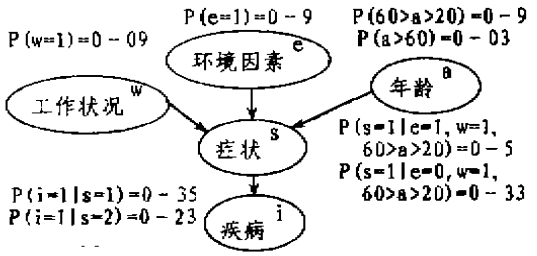

图1　用于描述疾病原因的贝叶斯网

在知识挖掘中应用贝叶斯网方法主要是根据样本数据来修正网络的结构和局部条件概率分布。这可以分为两种情况：(1)当网络结构已知，分析的目的是根据数据集合计算出每个弧所对应的后验条件概率分布。(2)当网络结构不确定时，分析的目的是能够根据数据集合从候选的网络结构中选择最合适的结构。无论是第一种情况还是第二种情况，都需在模型空间中搜索最优。其中的评价函数可以是：最大后验概率(MAP)，贝叶斯信息标准(BIC)，贝叶斯因子(BF)等。

贝叶斯网络应用于知识挖掘领域的优点是：抗噪音能力强，能够处理不完全的数据；可以发现隐藏在数据中的因果模型；可以很容易地结合领域知识，比如用概率来表达因果关系的强弱，先验的因果知识以候选贝叶斯网的形式出现。但其缺点在于：模型的复杂度随网络中节点的增加而呈指数级增长；通常只能找到较优解并且取决于所用的领域知识和启发性规则。

总之，统计模式识别方法具有良好的理论基础；描述模型的能力较强；可伸缩性、精确性、鲁棒性、抗噪音能力和开放性都较好；比较适合于数值信息。但大部分方法需要对概率分布作主观假设，因此需要主观知识支持；并且发现的结果多为数学公式，不易理解。

4 面向符号的机器学习方法

机器学习方法是采用人工智能技术来实现机器从客观世界中学习的能力。归纳学习是机器学习中最核

心、最成熟的分枝，旨在从大量的经验数据中归纳抽取一般的判定规则和模式。归纳学习可以发现分类、聚类、归纳描述等模式。归纳学习又可以根据有无导师信号分为有导师学习和无导师学习。

1）有导师学习　是事先由外部导师将训练例子分类，然后对这些带有导师信号的训练例子进行学习。有导师的学习也可以根据其学习策略分为以 AQ 系列为代表的覆盖算法和以 ID3为代表的分治算法。

覆盖算法的基本思想是：对于给定的正例集和反例集，在由属性构成的概念空间中，找到一个能够覆盖所有正例和排斥所有反例的最佳概念描述。所谓最佳即是在概念的充分完备性和概念的简明性上取得一致。最早的覆盖算法是 Michalski 的 AQ11算法，采用了自下而上的归纳方法，即在归纳过程中正例用于制导系统生成较概括的假设；反例用来减削不合适的假设。AQ11算法的特点在于采用约束星和 LEF 评价函数来约束对概念空间的搜索[4]。覆盖算法除了上述的自下而上的策略之外，还有自上而下和双向策略。

AQ11的后继算法在不同方面对算法作了改进，如 AQ15增加了构造性学习、渐进学习和近似推理的功能，而 AE 系列引进了扩张矩阵技术对降低算法的逻辑复杂度作了很大的改进[5]。覆盖方法是模拟人类的学习过程，因此所得出的知识较易为人理解，并且适合于符号数据，但因为需要对训练集作多遍扫描，大多数算法都是面向小训练集的。

分治方法的基本思想是用属性值对例子集合逐级划分形成树状结构，直到一个节点仅含有同一类的例子为止。分治算法的结果往往是一棵决策树。Quinlan 的 ID3算法是分治方法中的典型代表。在 ID3中使用了基于信息理论的启发式方法来选择属性，即根据每个属性的信息增益来衡量它对减少系统的不确定性的贡献[6]，以此来产生决策树。这样属性选择策略使得在剖分后的子树中对对象分类所要获得的信息最少。

在 ID3之后出现了许多改进算法，它们主要是从处理离散和连续量、处理大量数据、处理数据属性之间的关联关系使系统可利用非平行于轴的剖分面、处理非静态的训练集等方面进行了改进。如 ID4是一种递增式方法，通过不断获得的新信息更新决策树，适用于动态的训练集；C4.5可以同时处理具有离散和连续取值的属性并且使用信息增益比来选择属性使系统能够在分支数目和一次剖分后的分类准确度上进行权衡；CART-LC 和 OC1可以生成有斜剖分平面的决策树分类系统；SLIQ 则使用了一个特定的数据结构和 Gini 为分裂标准使算法更加适合于大型的训练集。

决策树方法的优点是适合于符号数据、知识易理解、算法精确并且鲁棒性强。缺点是对连续量数据分类能力较弱而且可能由于噪音的存在使得决策树过大。

2）无导师学习　此方法事先不知道训练例子的分类，算法通过学习能够对它们进行聚类。Michalski 等人的 CLUSTER/2的基本思想与前述的动态聚类类似，但两者在实现上有三点不同[4]：

- 对于距离的定义：在 CLUSTER/2中定义了对象之间的句法距离用于衡量名词性实体之间的相似性。
- 对于搜索空间的约束：在 CLUSTER/2中采用了约束星和 LEF 评价函数来实现对搜索空间的约束。
- 在 CLUSTER/2中，对所得到的聚类评价准则为：聚类和事件之间的适合度、聚类描述的简洁性、聚类之间的差异性和区分度等。

比较统计方法而言，概念聚类产生的结果比较容易理解，适合于名词性数据，但不适合于数值属性。同样概念聚类也需要对目标数据集进行多次扫描，因此不适合大训练集。

总之，面向机器学习的方法描述模型的能力较强；结果的可理解性、开放性较好；处理符号信息能力强。但它的精确性、鲁棒性和抗噪音能力一般，需要以算法的复杂度为代价，并且往往是面向经过整理过的小训练集，因此可伸缩性差。

5　神经网络方法

人工神经网络是一种典型的面向连接的学习技术。常用的人工神经网络有多种模型，每个模型有不同的倾向：如快速训练、快速联想和求最优解。就 KDD 而言，BP 网和 Kohonen 网应用最为广泛。

BP 网是由 Rumelhart 等人在1985年提出的，系统地解决了多层神经网络中隐层单元连接权的学习问题，并在数学上给出了完整的推导。BP 网采用多层前向的拓扑形状，由输入层、中间层和输出层组成。BP 网的学习策略为有导师学习，其学习过程由两部分组成：训练数据的正向传播和误差信号的反向传播，并且这种过程不断迭代，最后使得信号误差达到允许的范围之内。

BP 网能在最小均方差的意义下实现对样本输入输出对的逼近，它的学习能力强。与传统的统计分析相比，BP 网可以同时逼近多个输出；可以满足多输入参数所造成的复杂情况并且实现简单。BP 网可以被用于分类、回归和时间序列预测等 KDD 任务中。

Kohonen 网络是一个由全互连的神经元阵列形成的无导师自组织和自学习网络。它采用前向拓扑结构，其结构为两层神经单元：输入层和竞争输出层，并且两层之间全互连。Kohonen 网的基本思想是通过调整从

输入结点到竞争层上结点的连接权值，建立向量量化器。当训练开始时，连续取值且未指定预期输出的输入向量依次加载到网络上，竞争输出层的单元互相竞争，竞争胜利的单元表示与输入模式相对接近，于是该单元以及临近其他点的权值也被修正，使得这部分区域对相应输入敏感。随着训练的继续，该区域的半径越来越小，拥有最大权值的区域就对应着向量空间的聚类中心，此中心的点密度函数大致就是输入向量的概率密度函数，网络中的权值将被组织得使拓扑上接近的结点对物理上的相似输入敏感，因此 Kohonen 网是一种逐步形成特征映射的算法。

与其他聚类方法相比，Kohonen 网的优点在于：可以实现实时学习，网络具有自稳定性，无须外界给出评价函数，能够主动识别向量空间中最有意义的特征，抗噪音能力强等。

此外可用于 KDD 领域中的人工神经网络模型还可以有：ART 网、循环 BP 网(Recurrent Back Propagation)、RBF 网络(Radial Basis Function)和 PNN 网(Probabilistic Neural Networks)等。应能根据任务的特点来选择适当的网络模型。比如：发现的内容类别(如需要对时间序列预测时，用循环 BP 网要比普通的 BP 网较好)；训练数据的类型；训练数据量以及对训练速度的要求(如当需要在线学习时，ART 和 RBF 的训练数据相对而言要快)。

从 KDD 的角度来看，神经网络的模型描述能力强；精确性、鲁棒性和抗噪音能力都较好；一般不需要主观知识的支持。但它需要对数据做多遍扫描，训练时间较长，可伸缩性差；由于知识是以网络结构和连接权值的形式来表达的，因此结果的可理解性和开放性都很差。对于知识的可理解性，虽然可以通过观察输入和输出来分析网络内部的知识或者通过网络剪枝来简化网络的结构从而提取规则，但效果都不理想。

6 粗集理论和方法

粗集方法是近年来提出的用于对不完整数据进行分析、学习的方法。该方法与传统的统计分析和模糊集理论不同的是：后者需要依赖先验知识对不确定性的定量描述，如统计分析中的先验概率、模糊集理论中的模糊度等；而前者只依赖数据内部的知识，用数据之间的近似来表示知识的不确定性。下面先简单介绍粗集理论中的一些基本概念[7]：

定义决策系统 $DS=(U,C,D)$，其中 U 是实体集合，C 是条件属性集合，D 是决策属性集合，C 的任意一个子集 B 在 U 上定义了一个等价关系：

$$IND(B)=\{(x,y)\mid a(x)=a(y),\forall\ a\in B\}$$

该关系也称为由 B 定义的不可分辨关系。对于 $x\in U$、它的 B 等价类定义为：

$$[x]B=\{y\mid(x,y)\in IND(B)\}$$

对任意集合 X，R 是 U 的一个不可分辨关系，X 的 R 下近似和上近似分别定义为：

$$\underline{R}X=\{x\mid x\in U:[x]_R\subseteq X\};$$

$$\overline{R}X=\{x\mid x\in U:[x]_R\cap X\neq\varnothing\}$$

$\underline{R}X$ 表示根据关系 R 能够确定的被分入 X 的元素的集合，$\overline{R}X$ 表示根据关系 R 有可能被分入 X 的 U 的元素的集合。对于 $B\subset C$，定义 B 相对于 D 的正域：

$$POSB(D)=\bigcup\{\underline{B}X\mid X\in U\mid IND(D)\}$$

其中 $U\mid IND(D)$ 为 D 对 U 划分所得到的等价类集合。$POSB(D)$ 实际上是那些可以根据属性集合 B 准确地被分入由属性 D 所确定的分类的元素的集合。设 $a\in C$，若有 $POS_t(D)=POS_{c-\{a\}}(D)$，则称 a 为 C 中 D 可省略。当 C 中每个元素都不为 C 中 D 可省略的话，称 C 为 D 独立。当 $C'=C-C^*$ 为 D 独立，且 C^* 中的所有元素都是 D 可省略的话，则称 C' 为 C 的 D 相对简约。从分类的角度来看，相对简约就是用一种分类来表达另一种分类必不可少的属性集合。

D 和 C 之间的依赖程度可以由关系间的分类近似质量来度量：$r=|POS_t(D)|/|U|$。当 $r=1$ 表示全可导，$r<1$ 表示粗可导。属性 a 对于由决策属性 D 导致的分类的重要性就等于从 C 中删除 a 后 C 对 D 的依赖程度的变化值。

粗集方法在 KDD 中的应用可以是：

- 知识简约：简约和相对简约在粗集中是非常重要的概念，它反应了一个决策系统的本质。通过对条件属性集合的简约，可以保证简化后的决策系统具有与原先系统一样的分类能力。从数据预处理的角度看，属性简约能去掉多余属性，从而提高系统的效率。
- 属性相关分析：粗集方法中的属性重要程度可以用来衡量该属性对分类的影响程度，它与 ID3中的信息增益类似，可以证明两者在一定条件下是等价的[8]。
- 规则学习和决策表推导：在保证简化后的决策系统具有与原先系统一样的分类能力的前提条件下，通过使用知识简约和范畴简约，将决策系统简化并且找到最小(最短)决策规则集合，以达到最大限度泛化的目的。

总之，从 KDD 的角度来看，由于粗集方法中的决策表可以被视为关系型数据库中的关系表，因此粗集方法的伸缩性较强；鲁棒性和抗噪音能力都较强；知识的可理解性和开放性较好；比较适用于符号信息。此外，粗集方法可以对数据进行预处理，去掉多余属性，可提高发现效率，降低错误率。但是粗集方法的模型描述能力一般。

7 数据库技术

数据库技术是近年来发展得较为成熟的领域，如何利用已经相当成熟的数据库或数据仓库技术来为KDD服务是当前KDD研究领域中的一个热点。

1）OLAP和数据仓库　数据仓库是一种为了实现决策支持的数据模型、存储决策所需信息的语义一致的数据存储机制。数据仓库中的数据是经过清洗、整理过的数据，它能够向用户提供一致的、完整的数据集合。因此数据仓库为KDD的算法提供了一个很好的数据平台。

数据仓库的主要功能是提供联机分析过程(OLAP)[9]。OLAP提供的服务主要包括：切片和切块、概念上升、概念下钻、旋转等。OLAP是一种由用户驱动的、自上而下不断深入的分析工具。它有助于简化数据分析过程并且能够发现描述性的规则。但与KDD相比，OLAP更多地依赖于用户提出问题和指导并且它往往只用于对数据的总结描述。因此数据挖掘应用的广度和深度要大于OLAP。但数据仓库为数据挖掘提供了经过整理过的数据，同时OLAP的多角度、多层次的数据汇总能力为数据挖掘提供了良好的基础。

2）面向属性的归纳 AOI(Attribute-Oriented Induction)　该技术由Han提出[10]。该方法采用概念树爬升的技术来实现从关系表中归纳出高层次的总结性规则。AOI的基本思路是：首先得到与任务相关的以关系表形式出现的数据集合，然后通过对每个属性进行爬升概念树的方法对该属性进行泛化直到该属性中的不同值的个数小于一定的泛化阈值为止，最后合并数据集合中一致的元组。泛化后的关系中的每个泛化元组可以被视为一条对原关系的总结性特征描述规则。如果有另一个关系作为对照的话，对两个关系利用相同的办法，可以得到关系之间的区别描述规则。

AOI的优点在于过程简单、容易实现，并且在大型数据库上实现的效率较高，发现的知识易于理解，可以在多层次上发现知识，同时领域知识通过概念树的形式可以被容易结合进来。但它需要领域知识来构造每个属性对应的概念树并且定义泛化阈值，而且只能发现描述性知识。

总之，面向数据库的KDD方法的普遍优点在于：适合于在大型数据上的知识挖掘，伸缩性强；精确性、抗噪音和鲁棒性较好；对数值和符号数据都适合；知识的开放性和可理解性都较强。它的缺点在于：依赖数据模型，只能发现比较简单的描述性知识，用它来发现复杂模型较难。

总结　综上所述，KDD的各个方法的比较结果如表1所示。从表1可以看出，各个方法都有其优缺点和不同的适用领域，一个完整的知识发现系统应该能够根据实际需要充分地利用各方法的特点以达到扬长避短的目的。对不同方法的利用更多体现在知识发现过程的不同阶段，比如：

表1　知识发现各方法特点比较

	描述模型的能力	伸缩性	精确性	鲁棒性	抗噪音能力	知识可理解性	是否需要主观知识	开放性	适合的数据类型
统计模式识别方法	强	强	强	强	强	较差	需要	较差	数值
面向符号的机器学习方法	强	较差	一般	一般	一般	强	不需要	强	符号
神经网络方法	强	较差	强	强	强	较差	不需要	较差	皆可
粗集方法	一般	强	一般	强	强	强	不需要	强	符号
面向数据库的方法	较差	强	强	强	强	强	不需要	强	皆可

（下转第89页）

规则从前驱元素中演算得到。一般，这种"重写"关系及其对象逻辑的"项结构"允许"子项"的替换。很多定理证明器以这种风格的证明为基础，如著名的通用证明器 Isabelle，证明器 Affirm，Eves 等。定理证明器 Frgo[3]中采用的窗口推理证明技术也以此为基础。

使用基于"项重写"的定理证明器的最大好处在于其证明过程与精化过程相似，两者都包括选取待转换的成分，选择转换规则并进行实例化，验证规则的应用条件，以及用转换结果替换相应片段，两者都依赖相关对象和关系的相同性质：自反性，传递性，单调性。这可减轻用户在精化步骤和证明步骤转换时的负担。

窗口推理 PRT 使用基于"项重写"的窗口推理证明规范[4]，支持在层次证明结构中进行直接面向目标的推理。在任意阶段，证明可包含一层尚未解决的若干子问题及其相应上下文。在此层中的每个节点称为一个窗口。窗口内容包括：一个焦点，作为将要转换的项；一些假设，转换焦点时假定为真；一个关系，指出转换中保持的关系；一个目标，指出转换焦点所期望的结果。

利用这样的窗口，通过连续地在若干子问题上移动焦点完成该问题的证明。对于一个子问题，通过相应的假设，一焦点开启新的窗口。在这一窗口中，可以通过转换消解这一子问题，或者通过开启新窗口将其进一步分解。在一子问题解决之后，该窗口关闭，关闭的方法是：对于一个证明，将焦点还原为 TRUE，对于精化，则将规约转换为代码。这样，最终的证明可以表示为由若干窗口和转换组成的一种层次结构，由此构成了从最初问题到所需结果之间的正确转换。

使用这种规范，解决一个问题只需解决它的所有子问题。对于一个子问题，通过相应的假设，焦点开启新的窗口。在这个窗口内，可通过转换或分解进一步打开新的窗口处理问题。当子问题被解决，例如证明焦点为真或规约精化为代码，窗口将关闭。最终的证明表示为由众多窗口和转换组成的层次结构并一起构成了从初始问题到所求结果的正确转换。

在标准的定理证明器中，需要保持的关系一般为等价或蕴含，但别的关系也可能出现。当使用窗口推理进行精化转换时，是"被精化"关系。通过在前置条件下直接打开窗口并且弱化的操作可以替代象"弱化前置条件"这样的精化规则。

PRT 是通过对 Ergo 定理证明器扩展而建立的，使用窗口推理证明规范。Ergo 可扩充，通过具有理解力的策略语言支持自动证明，证明简单规则，无需用户干预，自动调用策略。

建立程序窗口推理逻辑 精化的标准处理以高阶逻辑为基础。程序变量作为域和其值的域明确区分。状态作为将变量映射到值的函数。表达式和谓词将状态分别映射到值和布尔值。程序起到谓词转换子的作用，而无需最弱前置条件算子。

精化演算的一个优点是大部分过程只需使用条件为一阶逻辑的精化规则，而且不需引用所有的状态。一些特别的逻辑条件用自然语言描述，在另外一些逻辑条件下，程序变量就象逻辑变量一样（例如，"$x=0\Rightarrow x\leqslant 1$"）。因为对所有的状态规则的逻辑条件一般隐式地表示为全称量词，所以可以缩小为一阶概念。

使用高阶方式描述精化，全称量词必须明确，各种标志符的表示也必须相互区分，作为表达式和谓词的参数的状态也不能随便隐藏，另外，这种方式使得形式化证明精化规则、清楚的推导状态和关系成为可能。使用没有清楚描述状态的一阶逻辑则不能做到。

程序窗口推理以模态逻辑为基础，这种模态逻辑与高阶逻辑语义有关。由于使用模态逻辑避免了高阶的结构，程序变量在规则的应用条件中可象模态逻辑的变量一样。使用约束状态集的模态算子可以区分标志符的不同出现。因为状态是谓词的参数，精化规则可以被证明。

程序窗口推理理论 Ergo 具有广泛的理论基础，覆盖了经典逻辑、集合论、代数和象序列这样的结构，为支持精化，使用程序窗口推理的理论被加进来。程序窗口推理理论提供对处理程序变量、应用精化规则和管理各种精化中的上下文的支持。它包括：

- 通过最弱前置条件函数(wp)得出的精化关系的定义。
- 作为理论中定理模式的精化规则。
- 应用于精化关系的窗口的打开和关闭。
- 处理程序变量和变量替换的机制。
- 应用精化规则的支持策略，对用户隐藏了许多细节并以简单的方式进行替换。
- 适合精化的证明界面。

每个精化步骤就是定理模式实例的应用，同时根据当前上下文对元变量进行相应实例化。Ergo 的理论认为最终程序是初始规约的精化结果。

程序变量 是有限常量集的模型。由于 Ergo 内嵌了不被模态逻辑使用的替换和量词的概念，不能直接使用逻辑变量。表达式替换程序变量表示为谓词的一个模态函数，由能计算可能发生替换的策略支持。类似地，可用全称和存在量词约束程序变量。

应用精化规则 在 Ergo 中，公理或定理被理解为直接推导规则。规则的应用通过实例化以匹配要使用的情形。其余模式变量暂时不被实例化。这些元变量可在以后推导的任意阶段实例化。对精化而言，应用规则时完全实例化更方便。所以每个精化规则都用一个参