监督学习的发展动态*)

蒋艳凰 周海芳 杨学军
(国防科学技术大学计算机学院 长沙 410073)

Current Directions in Supervised Learning Research

JIANG Yan-Huang ZHOU Hai-Fang YANG Xue-Jun
(School of Computers, National University of Defense Technology, Changsha 410073)

Abstract Supervised learning is very important in machine learning area. It has been making great progress in many directions. This article summarizes three of these directions, which are the hot problems in supervised learning field. These three directions are (a) improving classification accuracy by learning ensembles of classifiers, (b) methods for scaling up supervised learning algorithm, (c) extracting understandable rules from classifiers.

Keywords Supervised learning, Ensembles of classifiers, Rule extracting

1 引言

近10年来,机器学习领域的研究取得了突飞猛进的发展,首先表现在符号学习、计算学习理论、神经网络、统计方法、模式识别等的许多原来分离的研究团体,由于研究的进展与相互的交流走到了一起;其次,机器学习技术除了用于解决传统的学习问题,如语音识别、人脸识别、手写体汉字识别、医学数据分析、游戏等,还用于解决许多新问题,如数据库中的知识发现、语言处理、组合优化、机器人控制、遥感图像解译等。

监督学习是机器学习的重要方面,它可描述为:给定一些已知类别的样本$\{(X_1,y_1),(X_2,y_2),\cdots,(X_m,y_m)\}$,其中$X_i=(x_{i1},x_{i2},\cdots,x_{in})$为第$i$个样本的属性向量,元素$x_{ij}$为第$i$个样本中第$j$个属性的值,可以为离散或连续的值;$y_i\in\{1,2,\cdots,K\}$为第$i$样本的类别,属性向量$X$与类别$y$之间存在某种未知的复杂函数关系$y=f(X)$。利用监督学习算法对已知样本进行学习,得到一个分类器,用于近似该未知函数。分类器可以为决策树、神经网络、贝叶斯公式等多种形式。然后利用该分类器对未知类别的样本进行类别预测。在后面的讨论中,我们将学习得到的分类器用$h_1,h_2,\cdots,h_L$表示。

由于多个领域的融合,对监督学习产生了更多的需求,同时也出现了相应的解决方法。如对许多分类领域,分类精度十分重要,因此出现了多个分类器在分类结果或集成结构上的组合。对于信息量增加的今天,为了处理大规模问题,则需要对原有的学习算法进行扩展,从而加快学习速度,减少存储量。为了能够从已知的信息中挖掘出用户可理解的、有用的知识,提出了神经网络规则抽取、决策树剪枝等方法。本文选择了目前监督学习领域的3个热点问题进行讨论,即组合分类器、学习算法的扩展、提高算法的可理解性,介绍这3个方面的最新研究成果以及需要进一步解决的问题。

2 组合分类器

分类器组合是提高监督分类精度的有效手段,它通过某种方式将多个分类器组合起来,用于对未知样本分类。研究表明,只有当每个分类器的错误率小于0.5,且分类器之间的错误相互独立,对它们进行组合才能提高精度[1,2],图1给出了分类器个数为21,每个分类器的错误率均为0.3,且它们产生的错误之间相互独立时,恰有$n(0\leqslant n\leqslant 21)$个分类器分类错误的概率。可以看出,对于$n\geqslant 11$的所有情况,同时出错的概率之和仅为0.026,远远小于单个分类器的错误率。

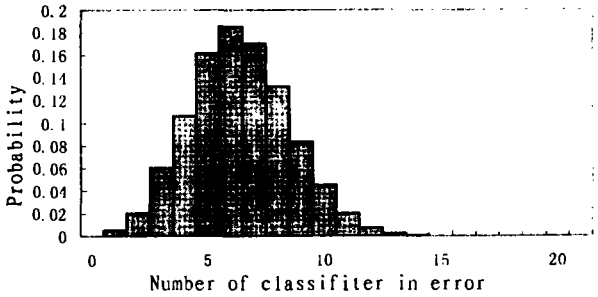


图1 恰有n个分类器出错的概率

组合分类器的研究包括2个方面:一是构造足够多的分类器,二是对这些分类器进行组合。

2.1 构建分类器

分类器的构建方法主要有以下几种:

①样本子集法。即对训练样本集进行操作,它采用同一个学习算法对不同的样本子集进行学习,从而生成多个分类器。这类方法对于那些不稳定(即训练集较小的改动会引起分类器大的变化)的算法,如决策树、神经网络、规则学习等方法,具有很好的效果。Bagging[3]、Boosting[4]及交叉验证[5]均属于这一类算法。

②属性选择法。即对训练样本的输入属性进行操作,当样本的属性很多时,每个分类器选择不同的属性进行学习,再对它们的结果进行综合。Cherkauer[6]利用这种方法对119种属性进行选择,生成4个不同结构的神经网络,并综合其结果,对金星上的火山进行识别。这种方法对于属性数目较少的情

*)本课题得到国家杰出青年科学基金资助(项目编号69825104)。蒋艳凰 博士研究生,主要研究方向为人工智能、机器学习、遥感图像处理等。周海芳 博士研究生,主要研究方向为图像处理、并行计算。杨学军,教授,博士生导师,研究领域为高性能计算。

况会降低分类精度[7]。

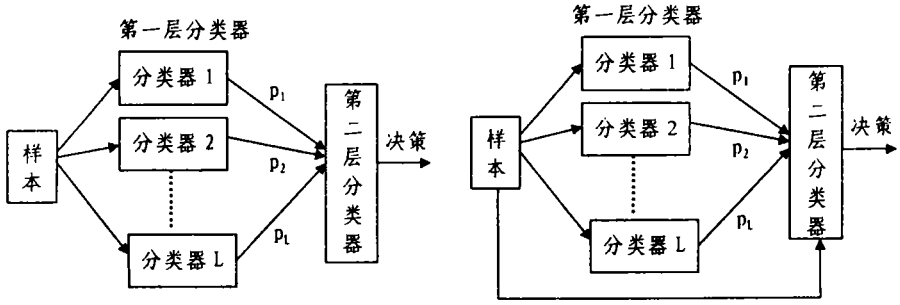
③类别编码法。Diretterich 和 Bakiri[8]提出错误校正输出编码法，该方法首先对类别进行编码，每个类别对应到一个长度为 L 的二进制位串，形成一个 $K\times L$ 的二维编码矩阵，K 为类别数；然后对编码矩阵的每一列进行学习，各得到一个二分器。对于编码串的第 $i(1\leqslant i\leqslant L)$ 位为 0 的类别，编码矩阵第 i 列产生的二分器对该类别的输出为 0，否则输出为 1；在分类时，每个二分器均对未知样本进行分类，得到一个码长为 L 的二进制位串。利用最小汉明距离法确定这一位串与哪个类别的编码最接近，从而获得该样本的类别。由于类别编码法可以将多类分类问题简化为两类分类问题，最近引起了研究者们的极大关注。Allwein[9]利用编码的方法给出了所有分类问题的一种通用表示；Crammer[10]分析了类别编码的学习能力，并设计出一种编码方法；Guruswami[11]将类别编码法与 Boosting 方法结合，进一步提高分类精度。

④随机因素法。在学习方法中注入随机因素。如 BP 神经网络的初始化权重是随机产生的，对于同样的样本集、同样的网络结构，采用不同的初始化权重，可以得到不同的神经网络分类器[12]。在决策树算法中，选择内部节点的扩展属性也是一种随机因素[13]。同一内部节点选择不同的扩展属性，可以得到不同的决策树。但实验结果表明，这种分类器生成方法不如样本子集法[5]。

⑤算法相关法。除了上述几种通用方法外，还可以根据具体算法生成不同的分类器。如 Opitz 和 Shavlik[14]提出利用遗传算法生成多个神经网络分类器，在每次迭代过程中，通过遗传算子选择网络拓扑结构，最终得到在精度与多样性方面满足要求的最佳的 N 个神经网络。Buntine[15]提出选择树(Option Tree)方法，这种决策树的内部节点可有多个扩展属性，每个扩展属性对应一棵子树。对样本进行分类时，每棵子树产生一个分类结果，最终利用投票的方法决策其属于哪一类。

2.2 分类器组合

得到多个分类器后，如何将它们的单个决策结果合并起来，形成最终的决策？一般有 3 种方法，最简单的方法是采用投票法，如果多数分类器的预测结果认为样本 X 属于第 i 类，则认为该样本属于第 i 类。对投票法的一种改进是采用概率估计值，由于分类器 h_l 得到的结果一般为每个类别的概率估计，而不是具体哪个类别，令分类器 h_l 预测样本 X 属于第 k 类的概率为：$P(f(X)=k|h_l)$，其中 $1\leqslant k\leqslant K$，则 L 个分类器的组合结果为：$P(f(X)=k)=\frac{1}{L}\sum_{l=1}^{L}P(f(X)=k|h_l)$，取组合概率估计最大的类别为样本 X 的类别。另一种方法是加权投票法，即参与投票时，每个分类器有一个权重，表示其结果占最终决策的份量，这种组合策略的关键在于确定各分类器的权重值。第三种方法称为分层决策法，每一层有多种分类器，后一层利用前一层的分类结果进行再次分类。分层决策法又有两种类型，一种是后一层仅利用前一层的预测结果作为输入[16]，另一种是后一层可利用其前面各层的预测结果以及初始样本作为输入[17]，图 2 给出了这两种方法的两层分层模型的结构。



(a) 分层模型 I　　(b) 分层模型 II

图 2 分层组合模型

组合分类器主要用于提高分类精度，但是也带来一系列问题：如存储量大，需要大量存储空间用于保存每个分类器；计算开销大，需要大量的计算时间用于每个分类器的学习；行为难以理解，即使单个分类器的行为可解释，易于理解，但多个分类器经过组合后，分类行为很难被用户理解。

3 学习算法的扩展

随着信息量的增加，人们需要对大量的已知样本进行学习。这就必须将原来的学习算法进行扩展，使之能够快速有效地处理上百个类别、上千个属性以及上万个样本的问题。如数据库知识挖掘问题，数据库每天完成成千上万的事务处理，人们希望能在几个小时内完成这些数据(样本)的分析任务。在信息检索领域，一个文档就是一个样本，文档中的每个字作为一个属性，则每个样本有上千个属性。在语音识别、目标识别、汉字识别等领域，则有众多的类别等待我们去决策。对于这些大规模问题，机器学习算法面临着处理时间长和存储量大的问题。

3.1 大规模样本的学习

目前对这类问题的研究主要集中在决策树算法的扩展上。一种方法是对样本集进行操作，如 Musick，Catlett 和 Rusell[18]提出的动态样本集选择法，可以根据决策的难易程度动态地选择训练样本集。对于离根节点近的内部节点，选择的样本集小。随着决策树的增长，逐渐扩大训练样本集。这种方法可以在不降低分类精度的情况下减少处理时间。

第二种方法是创建好的数据结构，用于存储训练样本，减少连续属性排序的时间。Shafer，Agrawal 和 Mehta[19]提出了 SPRINT 方法，它将原训练集分割成多个磁盘文件，每个属性一个文件。文件中的记录形式为 $\langle i,x_{i,j},y_i\rangle$，其 i 中表示训练样本序号，$x_{i,j}$ 表示该样本中第 j 个属性的值，y_i 为样本 i 的类别，文件中的记录按属性 j 的值从小到大排列；当选择属性 j 的分割点 θ 时，只需对属性 j 的文件进行顺序扫描。一旦获得 θ 的值，则将原属性 j 的文件分割为两个逻辑文件：一个包含第 j 个属性的值小于、等于 θ 的所有样本，一个包含属性 j 的值大于 θ 的样本，分别对应此内部节点的左右儿子节点。同时

建立 hash 表项，指明样本 i 属于新产生的分割节点哪一分枝；其它属性的磁盘文件也根据样本集的分布进行逻辑分割。SPRINT 算法并行性好，可以大大加快学习速度。Ruggieri[20] 对 C4.5 的具体实现进行改进，用二分法搜索和基数排序替代原来的线性搜索和快速排序算法，并加入 RainForest 内存管理方法，在结果相同的情况下大大加快 C4.5 决策树的生成速度。

第三种方法是对多个决策树进行组合[21]。它将样本集划分为 N 个互不相交的子集，对每个子集单独训练，分别得到一棵决策树，然后利用投票方式对未知样本进行分类。虽然每棵决策树的分类精度可能会降低，但通过组合可以提高精度。另外，这种方式可以并行生成决策树，加快了学习速度。

第四种方法是连续属性离散化处理。将连续属性离散成多个区间的形式，对区间进行处理，避免了对每个取值进行比较，加快运算速度。目前已有多种简单快速的方法用于连续属性的离散化，如 Catlett[22] 和 Fayyad[23] 采用多区间贪婪划分法（greedy multi-splitting），对连续属性的处理通过递归调用二分法（Binary Splitting）离散成多个区间，但理论证明这种多区间划分并不是最优划分。Fulton[24] 提出最优多区间（optimal multi-way partitions）划分法，并采用动态规划法寻找最优的 k 个区间。Elomaa 和 Rousu[25] 是在 Fulton 工作的基础上进行改进，引入边界点，加快离散化过程。

3.2 大规模属性的学习

许多问题领域都包含成百上千的属性，C4.5 和 BP 神经网络等算法难以直接应用。但从统计的角度看，很多属性都是无用的噪声属性，对它们的处理不仅浪费时间，而且导致生成的分类器精度低。在实际应用中，常常要对属性进行选择。

属性选择的方法有 3 种，一种是在学习前对数据进行初始化分析，选择一部分属性参与学习。最简单的方法是计算每个属性和各类别之间的互信息，如离散属性的互信息为：

$$\omega_j=\sum_{v}\sum_{c}P(y=c,x_j=v)\cdot\log\left(\frac{P(y=c,x_j=v)}{P(y=c)\cdot P(x_j=v)}\right)$$

对于连续属性，则将其离散化后再计算互信息。互信息代表了每个属性的重要程度。但这种方法独立地考虑每个属性，对于属性之间相关性大、必须利用多个属性的特殊组合才能进行分类的问题（如奇偶问题），互信息则起不到作用。为此，Kononenko[26] 提出了 RELIEF-F 算法，这种方法的基本思想是：随机抽取一些样本，计算它们的最近邻样本，然后调整各属性的权值，对分割属于不同类别的近邻样本能起到关键作用的属性获得的权重值大，并且证明对于相关属性问题，即使属性之间的相关性很大，该算法均能有效地给属性赋权重值[27]。

第二种方法是通过对学习算法的测试，选择效果最好的属性子集。如 John，Kohavi 和 Pfleger[28] 给出的 wrapper 算法，它每次选择一些属性，利用 10 次交叉验证的方法得到分类精度，然后选择分类精度最高的属性集，但这种方法时间复杂度很高。Moore 和 Lee[29] 结合 Leave-one-out 和最近邻方法，提出一种更为有效的方法，该方法每次从训练集中除去一个样本，并利用最近邻法对该样本进行分类。当处理完所有样本后得到的错误分类样本数为该学习算法的分类错误数。针对不同的属性集计算错误分类样本数，选择最好的属性集。

第三种方法是将属性选择融合到学习算法中。如 VSM（Variable-kernel Similarity Metric）方法[30] 就是在学习的过程中逐步确定高斯分布的参数以及各属性的权重。

4 提高算法的可理解性

数据挖掘就是从大型数据库中提取人们感兴趣的、有用的知识，监督学习算法在知识发现和数据挖掘领域有着广泛的应用。但是，一些监督学习算法，如 BP 神经网络，组合分类器等，可理解性很差。由于 BP 神经网络分类精度高，泛化能力强，应用广泛，本节主要针对如何提高 BP 神经网络的可理解性问题进行讨论。BP 神经网络学到的知识是通过网络连接权重表示出来，人们并不知道网络的输入与输出之间存在什么关系，哪个输入对哪个输出的影响更大，即存在所谓的“黑箱问题”，这限制了其在数据挖掘和知识发现等领域的应用。于是人们提出了从神经网络中抽取规则，将神经网络转化为决策树等方法来解决 BP 神经网络的黑箱问题。

4.1 基于结构分析的规则抽取

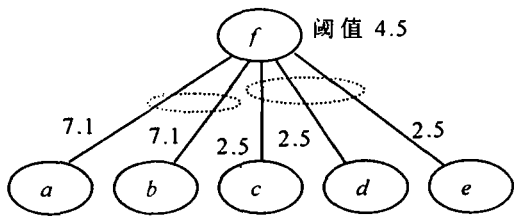
基于结构分析的神经网络规则抽取方法把规则抽取视为一个搜索过程，其基本思想是把已训练好的神经网络结构映射成对应的规则。由于搜索过程的计算复杂度和神经网络输入分量之间呈指数级关系，当输入分量很多时，会出现组合爆炸。因此，此类算法一般采用剪枝聚类等方法来减少网络中的连接，以降低计算复杂度。主要有如下几种算法：

Fu[31] 提出的 KT 算法将网络中结点的激活值近似处理为 0 和 1，将属性分为“正属性”和“负属性”，前者对某结论起到确认作用，后者则起否定作用。在所有的“负属性”都不出现的情况下，对所有最多由 k 个“正属性”组成的集合进行搜索，找出使某结论成立的最多有 k 个前件（相应于 k 个“正属性”）的规则，这些规则在“负属性”部分或全部出现的情况下，该结论仍然成立。虽然 KT 算法的规则搜索复杂度低，但抽取出的规则无法精确地描述原神经网络。

Towell 和 Shavlik[32] 提出基于知识的神经网络（Knowledge Based Artificial Neural Networks，KBANN），并从中抽取 MofN 规则。该算法先用标准聚类算法合并 KBANN 中权值接近的连接以创建等价类，并将每个等价类的权值设为该组连接权的平均值，然后去掉那些对结果影响不大的等价类，在不调整权值的前提下对神经网络重新进行训练，最后直接根据网络结构和权值抽取 MofN 规则，形式如下：

if (M of N antecedents are *true*) then …

MofN 规则形式不仅减少了抽取的规则数，还使得规则集比较简单易懂。另外，由于对连接进行了聚类，也使得规则搜索空间大为减少，从而较大地降低了规则抽取的时间开销。图 3 给出了一个典型的抽取 MofN 规则的例子：



虚线表示等价类

抽取的规则：if a then f;
if b then f;
if 2 of {c, d, e} then f.

图 3 MofN 算法规则抽取

Setiono[33] 提出了一种适用于三层前馈网络的通用型规则抽取算法。它使用激活值离散化技术和隐层神经元分裂技

术，递归地进行规则抽取处理。该算法可以产生相当精确的规则，但由于递归分裂要训练多个子网络，因此时间开销大，只适用于规模较小的网络。

为了避免组合爆炸的问题，大多数规则抽取算法都要对原神经网络进行剪枝操作，去掉一些不重要的连接。但为了保证神经网络的精度，需要对剪枝后的神经网络进行再训练。这增加了算法的开销，降低了算法的效率。为此，Setiono[34]又提出一种快速规则抽取算法——FERNN。该算法无需对神经网络进行多次训练，可以抽取 MofN 规则或是 DNF 规则。

4.2 基于性能分析的规则抽取

基于性能分析的神经网络规则抽取方法并不对神经网络结构进行分析和搜索，而是把神经网络作为一个整体来处理。这类方法注重的是抽取出的规则在功能上对网络的重现能力，即产生一组可以替代原网络的规则。

Saito 和 Nakano[35]提出的 RN 算法先从少数正例中抽取规则，然后根据未被覆盖的正例扩展规则，根据覆盖的反例缩减规则，直到规则覆盖了所有的正例，并且不覆盖任何反例为止。抽取的规则表示为 DNF 范式形式。由于该算法要搜索正反例空间，在示例空间较大时将面临组合爆炸问题。

Craven 和 Shavlik[36]为神经网络规则抽取任务下了一个定义：给定一个训练好的神经网络以及用于其训练的训练集，为网络产生一个简洁而精确的符号描述。显然，该定义来自于性能分析的角度，在该定义的基础上，将规则抽取视为一个目标概念为网络计算功能的学习任务。该算法使用了 2 个外部调用(Oracle)，其中 Examples的作用是为规则学习算法产生训练样本，Subset的作用则是判断被某个规则覆盖的样本是否都属于某个指定类。算法为每个分类产生各自的 DNF 表达式，它反复地通过 Examples产生训练样本。如果某训练样本没有被当前该类的 DNF 表达式覆盖，则新规则被初始化为该训练样本的所有属性值的合取。然后反复尝试去掉该规则的一些前件，并且调用 Subset来判断该规则是否与网络保持一致，从而使规则得以一般化。该算法不需使用特殊的网络训练方法，也不需将隐层神经元近似为阈值单元，但其计算量较大。

Thrun[37]为前馈神经网络提出了一种基于有效区间分析(Validity Interval Analysis)的规则抽取算法。该算法的关键是为所有或部分神经元找出激活区间，即有效区间。算法通过检查有效区间集合的一致性而不断排除导致不一致的区间。由于该算法的计算开销非常大，因此只适用于对规则进行理论验证，难以完成实际的神经网络规则抽取任务。

Benitez 等人[38]证明，多层前馈神经网络和基于模糊规则的系统是等价的，对任何一个以布尔函数为隐层神经元激活函数的单隐层前馈网络，均存在一个对应的模糊系统，使得网络可以由该系统的模糊规则进行解释。虽然由此出发可以很容易地获取一些模糊规则，但由于这些规则对不具有模糊数学背景的普通用户来说可理解性太差，因此在实际应用，尤其是数据挖掘任务中用处不大。

4.3 神经网络转化为决策树

决策树是一种可理解性高的分类方法，很容易转化为 IF-THEN 的规则形式。Boz[39]为了解决神经网络的黑箱问题，提出 DECTEXT 算法，直接将神经网络转化为决策树。这种算法把神经网络看成一个从输入到输出的映射函数，在不知道函数内部功能的情况下，寻找另一个相似的函数来代替该函数。DECTEXT 与传统决策树生成算法的不同点在于：①可以利用未知类别的样本生成决策树，样本的类别信息利用神经网络的输出来决定；②对连续属性的处理是将其离散成 $N+1$ 个区间，若某个内部节点选择了该连续属性作为扩展属性，则在该路径(从根节点到叶节点)上不再对该连续属性进行处理；③DECTEXT 算法没有停止标准(Stop Criterion)，当到达某个节点的所有样本被神经网络分为一类，则标记该节点为叶节点，类别为神经网络的输出类别；若到达某节点的所有样本被神经网络分为不同的类别，则递归调用 DECTEXT 算法；若到达某节点的样本数为零，则此节点为叶节点，并将该节点所在路径上的属性值作为神经网络的输入，输出的类别即为该节点的类别。

结论 除了上述 3 个方面，监督学习还有许多其它研究方向，如增强学习、简单贝叶斯方法和贝叶斯网络的研究、机器学习方法的过适应(overfitting)控制、算法的可理解性与分类精度之间的折衷，以及如何扩展监督学习的应用领域等，都是值得研究的课题。

参 考 文 献

1 Dietterich T. Machine learning research: four current directions. AI Magazine, 1997, 18(4): 97～136

2 Hansen L, Salamon P. Neural network ensembles. IEEE Trans. Pattern Analysis and Machine Intell, 1990, 12: 993～1001

3 Breiman L. Bagging predictors. Machine Learning, 1996, 24(2): 123～140

4 Freund Y, Schapire R E. Experiments with a new boosting algorithm. In Saitta, L. (Ed.), In: Proc. of the Thirteenth Intl. Conf. on Machine Learning, San Francisco, CA. Morgan Kaufman, 1996. 148～156

5 Parmanto B, Munro P W, Doyle H R. Improving committee diagnosis with resampling techniques. Advances in Neural Information Processing System. Cambridge, MA. MIT Press, 1996, 8: 882～888

6 Cherkauer K J. Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks. In Chan, Working Notes of the AAAI Workshop on Intergrating Multiple Learned Models, 1996. 15～21

7 Tumer K, Ghosh J. Error correlation and error reduction in ensemble classifiers. Connection Science, 1996, 8(3-4): 385～404

8 Dietterich T, Bakiri G. Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research, 1995, 2: 263～286

9 Allwein E, Schapire R E, Singer Y. Reducing multiclass to binary: A unifying approach for margin classifier. Journal of Machine Learning Research, 2000, 1: 113～141

10 Crammer K, Singer Y. On the learnability and design of output codes for multiclass problem. In: Proc. of the Thirteenth Annual Conf. on Computational Learning Theory, 2000. 35～46

11 Guruswanmi V, Sahai A. Multiclass learning, boosting, and error-correcting codes. In: Proc. of the Twelfth Annual Conf. on Computational Learning Theory. ACM press, 1999. 145～155

12 Kolen J F, Pollack J B. Back propagation is sensitive to initial conditions. In Advances in Neural Information Processing System. San Francisco, CA. Morgan Kaufmann, 1991, 3: 860～867

13 Kwok S W, Carter C. Multiple decision tree. Uncertainty in Artificial Intelligence, Elsevier Science, Amsterdam, 1990, 4: 327～335

14 Optiz D W, Shavlik J W. Generating accurate and diverse members of a neural-network ensemble. In Advances in Neural Infor-

mation Processing System. Cambridge, MA. MIT Press, 1996, 8:535～541
15 Buntine W L. A theory of learning classification rules: [Ph. D. thesis]. University of Technology, School of Computing Science, Sydney, Australia. 1990
16 Wolpert D. Stacked generalization. Neural Network, 1992, 5 (2):241～260
17 Gama J. Combining classification algorithm: [Ph. D. Thesis]. 1999
18 Musick R, Catlett J, Russell S. Decision theoretic subsampling for induction on large database. In: Utgoff, Proc. of the Tenth Intl. Conf. on Machine Learning. San Francisco, CA. Morgan Kaufmann, 1992. 212～219
19 Shafer J, Agrawal R, Mehta M. SPRINT: A scalable parallel classifier for data mining. In: Proc. of the Twenty-second VLDB Conf. San Francisco, CA. Morgan Kaufmann, 1996. 544～555
20 Ruggieri S. Efficient C4. 5. IEEE Transactions on Knowledge and Data Engineering, 2001, 14(2): 438～444
21 Chan P K, Stolfo S J. Learning arbiter and combiner trees from partitioned data for scaling machine learning. In: Proc. of the First Intl. Conf. on Knowledge Discovery and Data Mining. Menlo Park, CA. AAAI Press, 1995. 39～44
22 Catlett J. On changing continuous attributes into ordered discrete attributes. In: Proc. of the Fifth European Working Session on Learning, Heidelberg: Springer-Verlag, 1991. 164～178
23 Fayyad U M, Irani K B. Multi-interval discretization of continuous-valued attributes for classification learning. In: Proc. of the Thirteenth Intl. Joint Conf. on Artificial Intelligence. San Francisco, CA: Morgan Kaufman, 1993. 1022～1027
24 Fulton T, Kasif S, Salzberg S. Efficient algorithms for finding multi-way splits for decision trees. Machine Learning. In: Proc. of the Twelfth Intl. Conf. San Francisco, CA: Morgan Kaufmann, 1995. 244～251
25 Elomaa T, Rousu J. General and Efficient Multisplitting of Numerical Attributes. Machine Learning, 1999. 1～49
26 Kononenko I. Estimating attributes: analysis and extension of relief. In: Proc. of the 1994 European Conf. on Machine Learning. Amsterdan. Springer Verlag, 1994. 171～182
27 Kononenko I, Šimec E, Robnik-Šikonja M. Overcoming the myopic of inductive learning algorithms with RELIEFF. Applied Intelligence Journal, 1997, 7(1): 39～66
28 John G, Kohavi R, Pfleger K. Irrelevant features and the subset selection problem. In: Proc. of the eleventh Intl. Conf. on Machine Learning. San Francisco, CA. Morgan Kaufmann. 1994. 121～129
29 Moore A W, Lee M S. Efficient algorithm for minimizing cross validation error. In: Proc. of the eleventh Intl. Conf. on Machine Learning. San Francisco, CA. Morgan Kaufmann, 1994. 190～198
30 Lowe D G. Similarity metric learning for a variable-kernel classifier. Neural Computation, 1995, 7(1): 72～85
31 Fu L. Rule Learning by Searching on Adapted Nets. In: Proc. of the 9th National Conf. on Artificial Intelligence, Anaheim, CA: AAAI Press, 1991. 590～595
32 Towell G G, Shavlik J W. Knowledge-Based Artificial Neural Networks. Artificial Intelligence, 1994, 70(1-2): 119～165
33 Setiono R. Extracting Rules from Neural Networks by Pruning and Hidden-Unit Splitting. Neural Computation, 1997, 9(1): 205～225
34 Setiono R, Leow W K. FERNN: An algorithm for Fast Extraction of Rules from Neural Networks. Applied Intelligence, 2000, 12(1-2): 15～25
35 Saito K, Nakano R. Rule Extraction from Facts and Neural Networks. In: Proc. of the Intl. Neural Network Conf., North-Holland: Kluwer, 1990. 379～382
36 Craven M W. Extracting comprehensible models from trained neural network: [Ph. D thesis]. University of wisconsin, madison, 1996
37 Thrun S. Extracting Rules from Artificial Neural Networks with Distributed Representations. In: Tesauro G, Touretzky D, Leen T, eds. Advances in Neural Information Processing Systems (Volume 7), Cambridge, MA: MIT Press, 1995. 505～512
38 Benitez J M, Castro J L, Requena I. Are Artificial Neural Networks Black Boxes? IEEE Transactions on Neural Networks, 1997, 8(5): 1156～1164,
39 Boz O. Converting a trained neural network to a decision tree, DECTEX-Decision tree extractor. The 2002 Intl. conf. on machine learning and applications, 2002. 24～27

(上接第 3 页)

2 Bacchus F, Grove A, Halpern J Y, Koller D. From statistical knowledge bases to degrees of belief. Artificial intelligence, 1996, 87: 75～143
3 Boole G. The law of thought. Macmillan, London. 1854
4 Dekhtgar M I, Dekhtgar A, Subrahmanian V S. Hybird probabilistic programs: Algorithms and complexity. In: Proc. UAI-99, 1999. 160～169
5 Fagin R, Halpern J, Meggido N. A logic for reasoning about probabilities. Information and Computation, 1990, 87: 78～128
6 Gelfond M, Lifschitz V. The stable model semantics for logic programming. Logic programming. In: Proc. of the Fifth Int. Conf. and Symp. 1988. 1070～1080
7 Gelfond M, Lifschitz V. Classical negation in logic program and disjunctive databases. New generation computing, 1991. 365～387
8 Halpern J Y. An analysis of first-order logics of probability. Artificial Intelligence, 1990, 46(3): 311～350
9 Lloyd J. Foundations of logic programming. Springer, 1984
10 Lukasiewicz T. Probabilistic logic programming. In: Proc of the 13th Beinnial European Conf. On Artificial Intelligence, 1998. 388～392
11 Lukasiewicz T. Many-Valued Disjunctive Logic Programs with Probabilistic Semantics. In: Proc. of the 5th Intl. Conf. on Logic Programming and Nonmonotonic Reasoning, Volume 1730 of LNAI, 1999. 277～289
12 Lukasiewicz T. Probabilistic logic programming with conditional constraints. ACM Trans. Computat, Logic, 2001, 2(3): 289～337
13 Ng R, Subrahmanian V S. Probabilistic logic programming. Informathion and computation, 1992, 101(2): 150～201
14 Ng R, Subrahmanian V S. Stable semantics for probabilistic deductive databases. Information and computation, 1995, 110(1): 42～83
15 Ngo L, Haddawy P. Probabilistic Logic Programming and Bayesian Networks. ASIAN 1995, 1995. 286～300
16 Nilsson N J. Probabilistic logic. Artificial Intelligence, 1986, 28 (1): 71～87