

# 基于 Hebb 规则的分布神经网络学习算法

田大新<sup>1),2)</sup> 刘衍珩<sup>1),2)</sup> 李 宾<sup>3)</sup> 吴 静<sup>1),2)</sup>

<sup>1)</sup>(吉林大学计算机科学与技术学院 长春 130012)

<sup>2)</sup>(吉林大学符号计算与知识工程教育部重点实验室 长春 130012)

<sup>3)</sup>(吉林大学数学学院 长春 130012)

**摘 要** 随着知识发现与数据挖掘领域数据量的不断增加,为了处理大规模数据,scaling up 学习成为 KDD 的热点研究领域.文中提出了基于 Hebb 规则的分布式神经网络学习算法实现 scaling up 学习.为了提高学习速度,完整数据集被分割成不相交的子集并由独立的子神经网络来学习;通过对算法完整性及竞争 Hebb 学习的风险界的分析,采用增长和修剪策略避免分割学习降低算法的学习精度.对该算法的测试实验首先采用基准测试数据 circle-in-the-square 测试了其学习能力,并与 SVM,ARTMAP 和 BP 神经网络进行比较;然后采用 UCI 中的数据集 US-Census1990 测试其对大规模数据的学习性能.

**关键词** scaling up;数据分割;Hebb 规则;分布式学习;竞争学习

中图法分类号 TP183

## Distributed Neural Network Learning Algorithm Based on Hebb Rule

TIAN Da-Xin<sup>1),2)</sup> LIU Yan-Heng<sup>1),2)</sup> LI Bin<sup>3)</sup> WU Jing<sup>1),2)</sup>

<sup>1)</sup>(College of Computer Science and Technology, Jilin University, Changchun 130012)

<sup>2)</sup>(Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012)

<sup>3)</sup>(College of Mathematics, Jilin University, Changchun 130012)

**Abstract** In the fields of knowledge discovery and data mining the amount of data available for building classifiers or regression models is growing very fast. Therefore, there is a great need for scaling up inductive learning algorithms that are capable of handling very-large datasets and, simultaneously, being computationally efficient and scalable. In this paper a distributed neural network based on Hebb rule is presented to improve the speed and scalability of inductive learning. The speed is improved by doing the algorithm on disjoint subsets instead of the entire dataset. To avoid the accuracy being degraded as compared to running a single algorithm with the entire data, a growing and pruning policy is adopted, which is based on the analysis of completeness and risk bounds of competitive Hebb learning. In the experiments, the accuracy of the algorithm is tested on a small benchmark (circle-in-the-square) and compared with SVM, ARTMAP and BP neural network. The performance on the large dataset (USCensus1990Data) is evaluated on the data from UCI repository.

**Keywords** scaling up; data partition; Hebb rule; distributed learning; competitive learning

## 1 引 言

随着商业、政府、科研等领域信息数据的不断增

加,知识发现和数据挖掘(KDD)领域的科研人员通过对已有的机器学习、数据挖掘、模式识别等方法进行扩展以使其能够应用于大规模数据集,提出了 scaling up 归纳学习方法和许多实现技术. Scaling

up 学习方法关心的不仅仅是提高学习算法速度的问题,更关心的是扩展学习算法能否从大规模数据中有效地学习到知识.传统学习算法研究的重点是在有限(小规模)样本环境下如何使学习算法具有较好的推广(泛化)能力,面临的主要问题是过学习(overfitting)问题;而在大规模数据集下由于时间和空间的约束有可能无法对所有样本进行学习,从而产生了欠学习(underfitting)问题.研究人员提出的 scaling up 学习实现技术按大类分包括:设计快速的算法、利用关系表达和对数据进行分割等.快速算法的研究包括降低渐进复杂性、优化搜索和表示、利用问题自身的并行特征等<sup>[1-3]</sup>;关系表达主要研究的是充分利用知识的内在关系<sup>[4-5]</sup>;数据分割技术的研究包括将数据分割成子集、采样数据集、属性选择等<sup>[6-8]</sup>.

基于数据分割的 scaling up 归纳学习的主要过程是首先按分割规则  $R$  将完整数据集分割成子集;然后采用学习算法  $L$  对子集进行学习;最后采用合并机制  $C$  将学习结果组合得到最终的知识模型.在该领域研究人员通过对上述过程中  $R, L, C$  研究的侧重点不同,设计出了不同的 scaling up 学习算法.一类方法侧重对分割规则  $R$  的研究,其主要思想是从大规模数据集中采样出包含完整数据特征的小规模数据集,并对小规模数据进行学习从而在不改变传统学习算法的前提下实现对大规模数据的学习.为了降低采样对学习性能的影响,研究人员提出了递增采样机制,其对应的分割规则  $R$  是将样本  $D$  分割成  $D = \{D_0, D_1, \dots, D_n\}$ , 其中  $D_i < D_j (i < j)$ . 按子集增长比例的不同研究人员提出了算术采样<sup>[9]</sup>和几何采样<sup>[10]</sup>.为了判断采样子集多大时合适,通常采用学习曲线方法<sup>[11]</sup>.上述方法存在的主要问题是通过对采样的样本进行学习得到模型可能无法完整描述数据中暗含的知识,因为有可能部分知识包含在没有被采样的样本中.即使通过学习曲线方法能有效提高系统的准确率,但其有可能导致采样样本不断增加而使学习算法仍然面临大规模数据处理的难题.基于统计学的平衡样本数量和错误率的理论分析<sup>[12]</sup>以及增量学习算法<sup>[13-14]</sup>的研究是克服上述难题的重要途径.

另一类研究重点放在学习算法  $L$  和合并机制  $C$  上.这类 scaling up 归纳学习方法的分割规则  $R$  非常简单,只是将数据集  $D$  随机、均匀分割成  $n$  个互不相交的子集  $\{D_0, D_1, \dots, D_n\}$ , 每个子集  $D_i$  采用学习算法  $L_i$  对样本进行学习并通过合并机制  $C$  得到

最终描述系统的知识模型(规则集).各个子集对应的学习算法既可以相同也可以不同,如元学习算法采用不同的学习算法训练各个子集<sup>[15-16]</sup>.合并机制  $C$  解决的主要问题是如何将各个子集的学习结果合并起来组合成最终决策.除了常用的投票、加权投票法<sup>[17]</sup>外,元学习采用组合、仲裁法<sup>[18]</sup>, SCANN 方法<sup>[19]</sup>综合叠加<sup>[20]</sup>、一致性分析和最近邻方法.合并机制也是模块神经网络<sup>[21]</sup>、神经网络集成<sup>[22]</sup>、学习委员会机<sup>[23]</sup>研究的重点.此领域的研究出发点是将一个复杂任务分解成较简单的一系列子任务,每个子任务用一个神经网络(子模块)来完成,或是通过多个专家利用 Boosting/AdaBoost 方法提高系统的准确率.由于其学习过程仍然是集中的,因此如何将上述方法应用于大规模数据尚无有效机制.通过上述合并方法,即使各个子集的学习结果具有较大的误差,合并后的学习结果仍可以有效地提高系统的准确率.但是在如下情况下合并方法的性能与集中学习会有较大差距:①当存在大量冗余且学习结果并不精确的子集时;②当准确地预测或分类结果存在于某一子集中时.

利用相同或不同学习算法对互不相交子集分别进行学习并对学习结果进行组合的方法,既提高了对大数据集进行处理的速度又能保证学习结果的准确率.但目前用于解决大数据集问题的方法主要集中在决策树类学习算法<sup>[24-25]</sup>,这主要是因为决策树规则能够将各个子集得到的规则通过合并、剪枝等方法<sup>[26-27]</sup>将局部知识组合成全局知识,从而有利于避免上述两种造成合并方法产生较大误差情况的出现.模块神经网络、神经网络集成、学习委员会机等神经网络方法尽管其体系结构也是多个神经网络对样本进行学习,但现有方法无法有效地应用于大规模数据处理的原因包括:①其学习过程仍是集中式的,即所有或采样样本需提交给每个子神经网络(模块、专家),然后通过门网、投票等机制来将任务输入空间进行分割,从而提高系统的准确率;②神经网络是一种黑箱式系统,其知识存储在权重矩阵中,不同的神经网络学习算法的权重矩阵含义和应用方法不同,因此无法像合并决策树规则那样通过合并权重矩阵来组合各个子神经网络的学习结果.本文提出了基于 Hebb 规则的分布神经网络学习算法,让多个独立的神经网络同时处理随机分割的部分数据并将所得知识通过集中学习进行组合,这样在发挥单个神经网络并行处理能力的同时使其可以对分布存储的大规模数据集进行学习.

2 分布神经网络学习算法

神经网络的两个重要特征是分布和并行. 分布是指一个知识描述分布在多个处理节点中; 并行是指计算以并行的方式在分布的处理节点中进行. 尽管每个独立的神经网络以并行方式处理数据, 但让多个分布的神经网络合作处理一个任务则是一个难点. 因为神经网络的学习过程要求把所有的样本数据都提交给神经网络进行训练直至其在一次或多次循环训练后稳定. 这种机制使得当数据量非常大以至内存空间无法满足时学习无法进行. 本文提出的分布神经网络学习算法, 利用 Hebb 规则的局部学习特征, 通过增长、修剪机制实现对大规模数据的处理.

2.1 学习过程

学习算法的主要过程如图 1 所示, 其主要步骤如下.

- 1. 将大块数据集分割成小块, 然后将小块数据提交给各个独立的神经网络;
- 2. 各个神经网络对其分得的子数据集进行学习直至稳定;
- 3. 利用各个神经网络的学习结果生成新的数据集, 该数据集远小于各个子数据集之和;
- 4. 对新的数据集进行学习直至稳定.

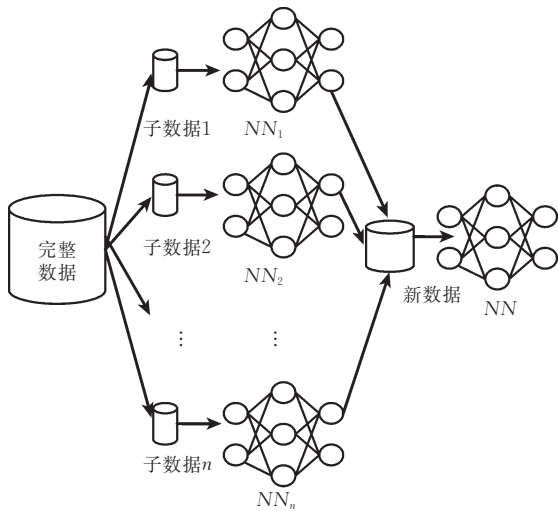


图 1 分布神经网络学习过程

每个子神经网络在接收第一个样本向量  $x_0$  后, 添加第一个神经元  $w_0$  并置初始值为  $x_0$ ; 在后续提交样本进行训练时, 首先判断学习是否结束, 若否, 则计算相似度值, 然后由竞争函数计算出竞争获胜的神经元; 若获胜神经元的相似度小于等于相似度门限值, 则根据学习算法修正神经元权重值, 否则添加新的神经元并将新添加的神经元的权重值置为该次

提交的样本向量  $x_i$ .

整个过程包括两个学习阶段: 分布学习和集中学习. 分布学习的样本为子数据集, 学习算法既要保证学习到子数据集中的完整知识, 又不能丢弃因分割产生的部分知识; 集中学习的样本为由各个子神经网络学习结果组成的新数据集, 该数据集不仅包含各子数据集的知识而且样本数量远小于原始数据集. 一个稳定的 Hebb 神经网络学习到的知识存储在权重矩阵  $W_{(m \times n)}$  中, 其中  $m$  为神经元的个数,  $n$  为每个神经元的维数. 在上述学习过程中每个分布学习的神经网络中神经元的维数  $n$  等于每个样本  $x_{(1 \times n)}$  的维数, 当神经网络稳定后, 其权重矩阵  $W$  的每一行为从子数据集中学习得到的知识. 例如, 原始数据集  $X$  含有  $p \times q$  个样本, 数据集  $X$  被分割成  $p$  个子数据集  $^{(i)}X$  ( $i=1, 2, \dots, p$ ), 每个子数据集含有  $q$  个样本. 当对  $^{(i)}X$  进行学习的神经网络学习稳定后其权重矩阵  $^{(i)}W$  含有  $^{(i)}r$  行 ( $^{(i)}r \ll q$ ), 由所有  $^{(i)}W$  中的知识点生成新的样本数据集  $\tilde{X}$  远小于原始数据集  $X$ .

2.2 Hebb 规则

上述学习过程可以通过基于 Hebb 规则类神经网络学习算法实施是因为该类神经网络具有如下两个特征: ① Hebb 学习是一种局部学习; ② 该类神经网络的权重向量代表了知识点. 这两个特征确保了即使代表某类知识的训练样本被分割到多个子集中也能在分布学习时被保留并在集中学习后被抽取出来, 从而避免了当存在大量冗余且学习结果并不精确的子集时分割学习误差较大的问题. 而其它学习算法因无法同时具备上述特征而无法采用上述学习过程对大规模数据进行学习. 如 BP 类学习算法是一种全局优化过程, 因此分布学习过程中各子 BP 神经网络会尽最大可能对子集进行学习, 这种方式可能会丢弃包含在子集中的不完整的知识点; 此外, BP 神经网络的学习结果存储于权重矩阵中, 权重矩阵对外界来说是一个黑盒, 其可用的信息只是针对某一输入各子 BP 网络给出的分类或回归结果, 因此集中学习过程只能采用投票类方法得出最优的结论, 而无法通过集中学习对学习结果进一步学习从而避免子集学习准确率低或某一结论只存在于某一子集的问题. RBF 类学习算法具备第一类特征, 但其第二层网络的权重矩阵也是一个黑盒, 因此集中学习时仍面临与 BP 网络同样的问题.

根据 Hebbian 假设, 可用能量函数表示一个 Hebb 神经元的学习规则

$$E(\mathbf{w}) = -\phi(\mathbf{w}^T \mathbf{x}) + \frac{\alpha}{2} \|\mathbf{w}\|_2^2 \quad (1)$$

其中,  $\mathbf{w}$  是突触权重向量,  $\mathbf{x}$  是输入样本向量,  $\phi(\cdot)$  为可微函数,  $\alpha \geq 0$  为遗忘系数. 神经元的输出为

$$y = \frac{d\phi(v)}{dv} = f(v) \quad (2)$$

其中,  $v = \mathbf{w}^T \mathbf{x}$  是神经元的活跃系数. 通过快速下降法导出连续时间的学习规则

$$\frac{d\mathbf{w}}{dt} = -\mu \nabla_{\mathbf{w}} E(\mathbf{w}) \quad (3)$$

其中,  $\mu > 0$  为学习速度系数,  $\nabla_{\mathbf{w}} E(\mathbf{w}) = \partial E(\mathbf{w}) / \partial \mathbf{w}$ , 则式(1)的梯度为

$$\begin{aligned} \nabla_{\mathbf{w}} E(\mathbf{w}) &= -f(v) \frac{\partial v}{\partial \mathbf{w}} + \alpha \mathbf{w} \\ &= -y \mathbf{x} + \alpha \mathbf{w} \end{aligned} \quad (4)$$

由此可得单神经元的学习规则为

$$\frac{d\mathbf{w}}{dt} = \mu [y \mathbf{x} - \alpha \mathbf{w}] \quad (5)$$

则离散时间的学习规则为

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \mu [y(t+1) \mathbf{x}(t+1) - \alpha \mathbf{w}(t)] \quad (6)$$

在竞争学习中, 神经网络中的输出神经元彼此通过竞争来成为活跃的, 正是这个特性使竞争学习适合于发现统计上的突出特征. 传统的竞争机制为胜者全得, 即每次只有一个神经元是激活的, 在基于 Hebb 规则的神经网络里, 除采用胜者全得机制的 ART<sup>[28]</sup>、PCA<sup>[29]</sup> 等外, 也有若干输出神经元同时处于激活状态的 SOM<sup>[30]</sup>、RPCL<sup>[31]</sup> 等. 为了克服竞争中竞争层神经元个数固定导致无法适用于事先不知道类别数目、数据提交的顺序和学习速度等参数的选择会导致类别中心来回振荡以及神经元数目过多从而导致神经网络过度拟合数据等问题. 除了上述经典神经网络采用基于 Hebb 规则的竞争学习算法外, 还有很多类似的学习算法, 如 RPCL<sup>[31]</sup> 的思想是在每次学习中, 与输入最为相似的神经元得到学习, 同时对第二相似的进行惩罚, 使其中心远离输入样本. DGNN<sup>[32]</sup>、LTCL<sup>[33]</sup> 对所有神经元根据竞争的结果实施不同级别的奖励和惩罚.

### 2.3 完整性分析

神经网络学习算法是一个从预测函数集  $\{L(y, f_w(\mathbf{x}))\}$  中选择一个适当的函数  $f_w^*(\mathbf{x})$ , 使预测期望风险

$$R(f_w(\mathbf{x})) = \int L(y, f_w(\mathbf{x})) dp(\mathbf{x}, y) \quad (7)$$

最小的过程, 其中  $L(y, f_w(\mathbf{x}))$  为由于用  $f_w(\mathbf{x})$  对  $y$

进行预测造成的损失. 通常概率分布  $p(\mathbf{x}, y)$  是未知的, 无法直接最小化风险泛函, 但得到了依  $d(\mathbf{x}_i, \mathbf{W}_j) \leq \lambda$  独立地随机抽取出的观测样本

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n) \quad (8)$$

因此采用经验风险泛函

$$R_{\text{emp}}(f_w(\mathbf{x})) = \frac{1}{n} \sum_{i=1}^n L(y_i, f_w(\mathbf{x}_i)) \quad (9)$$

来逼近式(7)定义的风险泛函.

分布式学习算法对随机分割的数据进行学习的结果如果等价于对完整数据的学习结果, 则说明分布式学习算法具备完整性. 从式(8)中的有限数据点恢复其背后隐含的函数  $f(\mathbf{x}, \mathbf{w})$  是一个反问题, 因而往往是不适定的, 为此 Tikhonov 提出了正则化方法解决不适定问题. 正则化理论要求的最小化泛函为

$$R(\mathbf{w}) = R_s(\mathbf{w}) + \lambda R_c(\mathbf{w}) \quad (10)$$

其中  $R_s(\mathbf{w})$  为实际风险项,  $R_c(\mathbf{w})$  为正则化项,  $\lambda$  为正则化参数. 正则化的基本思想是通过某些含有解的先验知识的非负的辅助泛函来使解稳定. 分析结果表明, 本文提出的分布式学习方法与采用正则化理论的学习方法等价.

**定理 1.** 基于 Hebb 规则的分布神经网络学习算法等价于正则化方法.

证明. 分布学习得到的权重矩阵<sup>(i)</sup> $\mathbf{W}$  的每一个行向量<sup>(i)</sup> $\mathbf{W}_j$  为一部分样本的邻域函数中心, 因此对于该部分样本中的一个样本<sup>(i)</sup> $\mathbf{X}_k$ ,

$$^{(i)}\mathbf{W}_j = ^{(i)}\mathbf{X}_k + \mathbf{A}_k \quad (11)$$

在新数据集<sup>(i)</sup> $\tilde{\mathbf{X}}$  中,

$$^{(i)}\tilde{\mathbf{X}}_l = ^{(i)}\mathbf{W}_j + \mathbf{B}_m \quad (12)$$

其中  $|\mathbf{A}_{ki}| \leq \beta$ ,  $|\mathbf{B}_{mi}| \leq \beta$  合并式(11)和式(12)得到

$$^{(i)}\tilde{\mathbf{X}}_l = ^{(i)}\mathbf{X}_k + \mathbf{A}_k + \mathbf{B}_m \quad (13)$$

对于局部风险最小化模型中的误差函数  $L(y, f_w(\mathbf{x}))$  在新数据集<sup>(i)</sup> $\tilde{\mathbf{X}}$  中为  $L(y, f_w(\mathbf{x} + \mathbf{a} + \mathbf{b}))$ , 将  $f_w(\mathbf{x} + \mathbf{a} + \mathbf{b})$  按泰勒级数展开得

$$\begin{aligned} f_w(\mathbf{x} + \mathbf{a} + \mathbf{b}) &= f_w(\mathbf{x}) + \nabla f_w(\mathbf{x} + \mathbf{a} + \mathbf{b})^T (\mathbf{a} + \mathbf{b}) + \\ &\quad \frac{1}{2} (\mathbf{a} + \mathbf{b})^T \nabla^2 f_w(\mathbf{x} + \mathbf{a} + \mathbf{b}) (\mathbf{a} + \mathbf{b}) + \dots \\ &= f_w(\mathbf{x}) + \Delta g(\mathbf{x}) \end{aligned} \quad (14)$$

其中  $|\mathbf{a}_i + \mathbf{b}_i| \leq 2\beta$ ,  $\nabla f(\mathbf{z})$  为  $n$  元函数  $f(\mathbf{z}) = f(z_1, z_2, \dots, z_n)$  的梯度,  $\nabla^2 f(\mathbf{z})$  为赫森矩阵.  $L(y, f_w(\mathbf{x}))$  取最小二乘法, 则

$$\begin{aligned} R(\mathbf{w}) &= \int (y - f_w(\mathbf{x} + \mathbf{a} + \mathbf{b}))^2 dp(\mathbf{x}, y) \\ &= \int (y - f_w(\mathbf{x}))^2 dp(\mathbf{x}, y) + \end{aligned}$$

$$\int ((y - f_w(\mathbf{x}))\Delta g(\mathbf{x}) + \Delta g(\mathbf{x})^2) d\rho(\mathbf{x}, y) \quad (15)$$

由式(15)可以发现分布式学习即为在风险泛函  $\int (y - f_w(\mathbf{x}))^2 d\rho(\mathbf{x}, y)$  的基础上增加惩罚项  $\int ((y - f_w(\mathbf{x}))\Delta g(\mathbf{x}) + \Delta g(\mathbf{x})^2) d\rho(\mathbf{x}, y)$ , 这种方法在均衡神经网络的偏置与方差<sup>[34]</sup>中普遍采用, 因此在  $\beta$  控制在一定小的范围内<sup>[35]</sup>等价于正则化方法。证毕。

## 2.4 局部学习风险界分析

通过分析上述 Hebb 规则可以发现其特征是首先通过竞争选出与样本  $\mathbf{x}_i$  距离在一定范围内的神经元  $\mathbf{W}_j$ , 即  $d(\mathbf{x}_i, \mathbf{W}_j) \leq \lambda$ , 然后按着 Hebb 规则修正  $\mathbf{W}_j$  的值。这种学习过程的本质是不同  $\mathbf{W}_j$  对其周围的样本最小化风险, 所以该学习过程是一种局部学习<sup>[36]</sup>, 但与文献<sup>[37]</sup>中定义的局部风险最小化模型不同的是邻域中心为竞争获胜的预测函数  $L(y, f_w^*(\mathbf{x}))$ , 因此本文定义如下最小化风险模型。

**定义 1.** 竞争函数  $C(\mathbf{x}, \mathbf{W}, \lambda)$ , 对于 ART 类学习算法

$$C(\mathbf{x}, \mathbf{W}, \lambda) = \begin{cases} 1, & d(\mathbf{x}, \mathbf{W}_j) \leq \lambda \\ 0, & \text{其它} \end{cases} \quad (16)$$

当  $d(\mathbf{x}, \mathbf{W}_j) > \lambda$  时谐振发生; 对于 SOM 类学习算法

$$C(\mathbf{x}, \mathbf{W}, \lambda) = \exp\left\{-\frac{d(\mathbf{x}, \mathbf{W}_j)}{\lambda}\right\} \quad (17)$$

在典型 SOM 学习中  $\lambda$  随学习过程逐渐缩小。

**定义 2.** 基于 Hebb 规则的竞争学习算法的风险泛函

$$RC(f_w(\mathbf{x}), \lambda) = \int C(\mathbf{x}, \mathbf{W}, \lambda) L(y, f_w(\mathbf{x})) d\rho(\mathbf{x}, y) \quad (18)$$

**定义 3.** 基于 Hebb 规则的竞争学习算法的经验风险泛函

$$RC_{\text{emp}}(f_w(\mathbf{x}), \lambda) = \frac{1}{n} \sum_{i=1}^n C(\mathbf{x}_i, \mathbf{W}, \lambda) L(y, f_w(\mathbf{x}_i)) \quad (19)$$

为了估计经验风险最小化的推广能力和学习过程的收敛速度, 按照统计学习理论需估计风险泛函  $RC(f_w(\mathbf{x}), \lambda)$  所能达到的风险值和这一风险值接近于最小可能风险值的程度。

**定理 2.** 包含  $r$  个有限元素的函数集  $\{C(\mathbf{x}, \mathbf{W}, \lambda) L(y, f_w(\mathbf{x}))\}$  和随机抽取出的  $n$  个观测样本, 对于最小化经验风险  $RC_{\text{emp}}(f_w(\mathbf{x}), \lambda)$  的函数  $(f_w^a(\mathbf{x}), \lambda^a)$  不等式

$$RC(f_w^a(\mathbf{x}), \lambda^a) \leq RC_{\text{emp}}(f_w^a(\mathbf{x}), \lambda^a) + \sqrt{\frac{\ln 2r - \ln \mu}{2n}} \quad (20)$$

依至少  $1 - \mu$  的概率成立。

证明. 由 Glivenko-Cantelli 定理可知对于任何给定的概率测度  $P$  和任何给定的  $\beta > 0$ , 则

$$P\{\sup_{n \rightarrow \infty} |P(A) - v_n(A)| > \beta\} \rightarrow 0 \quad (21)$$

其中  $v_n(A)$  为在  $n$  次独立随机试验中事件  $A$  出现的频率。Glivenko-Cantelli 定理说明当试验次数  $n$  趋于无穷大时频率收敛于概率。Chernoff 不等式

$$P\{\sup(P(A) - v_n(A)) > \beta\} \leq 2\exp\{-2\beta^2 n\} \quad (22)$$

$$P\{\sup(v_n(A) - P(A)) > \beta\} \leq 2\exp\{-2\beta^2 n\} \quad (23)$$

给出了收敛速率。对于指示函数风险泛函  $RC(f_w(\mathbf{x}), \lambda)$  定义了概率, 经验泛函  $RC_{\text{emp}}(f_w(\mathbf{x}), \lambda)$  定义了频率, 所以根据式(22)可以得到

$$\begin{aligned} P\{\sup_{1 \leq j \leq r} (RC(f_w^j(\mathbf{x}), \lambda^j) - RC_{\text{emp}}(f_w^j(\mathbf{x}), \lambda^j)) > \beta\} \\ \leq \sum_{j=1}^r P\{(RC(f_w^j(\mathbf{x}), \lambda^j) - RC_{\text{emp}}(f_w^j(\mathbf{x}), \lambda^j)) > \beta\} \\ \leq 2r\exp\{-2\beta^2 n\} \end{aligned} \quad (24)$$

若定义

$$\mu = 2r\exp\{-2\beta^2 n\} \quad (25)$$

则求出

$$\beta = \sqrt{\frac{\ln 2r - \ln \mu}{2n}} \quad (26)$$

由不等式(24)可得对于函数集  $\{C(\mathbf{x}, \mathbf{W}, \lambda) L(y, f_w(\mathbf{x}))\}$  中的所有  $r$  个函数, 不等式

$$RC(f_w^j(\mathbf{x}), \lambda^j) - RC_{\text{emp}}(f_w^j(\mathbf{x}), \lambda^j) \leq \beta \quad (27)$$

依  $1 - \mu$  的概率成立。因此对于最小化经验风险  $RC_{\text{emp}}(f_w(\mathbf{x}), \lambda)$  的函数  $(f_w^a(\mathbf{x}), \lambda^a)$ , 不等式(27)同样成立, 将式(26)代入不等式(27)得到不等式(20)依至少  $1 - \mu$  的概率成立。证毕。

对于最小化风险  $RC(f_w(\mathbf{x}), \lambda)$  的函数  $(f_w^\varepsilon(\mathbf{x}), \lambda^\varepsilon)$ , 由不等式(23)可得

$$\begin{aligned} P\{(RC_{\text{emp}}(f_w^\varepsilon(\mathbf{x}), \lambda^\varepsilon) - RC(f_w^\varepsilon(\mathbf{x}), \lambda^\varepsilon)) > \beta_1\} \leq \\ 2\exp\{-2\beta_1^2 n\} \end{aligned} \quad (28)$$

令  $2\exp\{-2\beta_1^2 n\} = \mu$ , 则

$$\beta_1 = \sqrt{\frac{\ln 2 - \ln \mu}{2n}} \quad (29)$$

由不等式(28)可得

$$RC(f_w^\varepsilon(\mathbf{x}), \lambda^\varepsilon) \geq RC_{\text{emp}}(f_w^\varepsilon(\mathbf{x}), \lambda^\varepsilon) - \sqrt{\frac{\ln 2 - \ln \mu}{2n}} \quad (30)$$

依  $1-\mu$  的概率成立. 因为  $(f_w^a(x), \lambda^a)$  最小化经验风险  $RC_{\text{emp}}(f_w(x), \lambda)$ , 所以

$$RC_{\text{emp}}(f_w^\epsilon(x), \lambda^\epsilon) \geq RC_{\text{emp}}(f_w^a(x), \lambda^a) \quad (31)$$

由定理 2 和不等式 (30), (31) 可推出不等式

$$RC(f_w^a(x), \lambda^a) - RC(f_w^\epsilon(x), \lambda^\epsilon) \leq \sqrt{\frac{\ln 2r - \ln \mu}{2n}} + \sqrt{\frac{\ln 2 - \ln \mu}{2n}} \quad (32)$$

依至少  $1-2\mu$  的概率成立.

对于包含无穷多个元素的函数集  $\{L(y, f_w(x))\}$ , Vapnik 证明了对于随机抽取出的  $2n$  个观测样本和任何给定的  $\beta > 0$

$$P\left\{\sup\left(\frac{R(f_w(x)) - R_{\text{emp}}(f_w(x))}{\sqrt{R(f_w(x))}}\right) > \beta\right\} \leq 4\exp\left\{\left(\frac{H_{\text{ann}}(2n)}{n} - \frac{\beta^2}{4}\right)n\right\} \quad (33)$$

$H_{\text{ann}}(n)$  为指示函数集在大小为  $n$  的样本集上的退火熵

$$H_{\text{ann}}(n) = \ln EN((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)) \quad (34)$$

$N((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$  为函数集  $\{f_w(x)\}$  分离样本的分法数目,  $E$  为  $N$  的期望值.

由于  $RC(f_w(x), \lambda)$  是由两个函数集  $C(x, W, \lambda)$  和  $L(y, f_w(x))$  共同决定的, 若  $C(x, W, \lambda)$  的退火熵为  $H_{\text{ann}}^C$ ,  $L(y, f_w(x))$  的退火熵为  $H_{\text{ann}}^L$ , 则  $C(x, W, \lambda)L(y, f_w(x))$  的退火熵  $H_{\text{ann}}^{CL} \leq H_{\text{ann}}^C + H_{\text{ann}}^L$ .

**定理 3.** 包含无穷多个元素的函数集  $\{C(x, W, \lambda)L(y, f_w(x))\}$  和随机抽取出的  $2n$  个观测样本, 对于最小化经验风险  $RC_{\text{emp}}(f_w(x), \lambda)$  的函数  $(f_w^a(x), \lambda^a)$  不等式

$$RC(f_w^a(x), \lambda^a) \leq RC_{\text{emp}}(f_w^a(x), \lambda^a) + \frac{2H_\mu}{n} + 2\sqrt{\frac{H_\mu}{n}\left(RC_{\text{emp}}(f_w^a(x), \lambda^a) + \frac{H_\mu}{n}\right)} \quad (35)$$

依至少  $1-\mu$  的概率成立, 其中  $H_\mu = H_{\text{ann}}^C(2n) + H_{\text{ann}}^L(2n) + \ln 4 - \ln \mu$ .

证明. 由不等式 (33) 可得

$$\begin{aligned} P\left\{\sup\left(\frac{RC(f_w(x), \lambda) - RC_{\text{emp}}(f_w(x), \lambda)}{\sqrt{RC(f_w(x), \lambda)}}\right) > \beta\right\} \\ \leq 4\exp\left\{\left(\frac{H_{\text{ann}}^{CL}(2n)}{n} - \frac{\beta^2}{4}\right)n\right\} \\ \leq 4\exp\left\{\left(\frac{H_{\text{ann}}^C(2n) + H_{\text{ann}}^L(2n)}{n} - \frac{\beta^2}{4}\right)n\right\} \quad (36) \end{aligned}$$

令  $4\exp\left\{\left(\frac{H_{\text{ann}}^C(2n) + H_{\text{ann}}^L(2n)}{n} - \frac{\beta^2}{4}\right)n\right\} = \mu$ , 则

$$\beta^2 = \frac{4}{n}(H_{\text{ann}}^C(2n) + H_{\text{ann}}^L(2n) + \ln 4 - \ln \mu) = \frac{4H_\mu}{n} \quad (37)$$

由不等式 (36) 可推出对于函数集中的所有函数, 不等式

$$\frac{RC(f_w(x), \lambda) - RC_{\text{emp}}(f_w(x), \lambda)}{\sqrt{RC(f_w(x), \lambda)}} \leq \beta \quad (38)$$

依至少  $1-\mu$  的概率成立, 由不等式 (38) 可得

$$(RC(f_w(x), \lambda))^2 - (2RC_{\text{emp}}(f_w(x), \lambda) + \beta^2) \cdot$$

$$RC(f_w(x), \lambda) + (RC_{\text{emp}}(f_w(x), \lambda))^2 \leq 0 \quad (39)$$

要使不等式 (39) 成立,  $RC(f_w(x), \lambda)$  应同时满足如下条件:

$$RC(f_w(x), \lambda) \geq \frac{\beta^2 - \beta \sqrt{4RC_{\text{emp}}(f_w(x), \lambda) + \beta^2}}{2} + RC_{\text{emp}}(f_w(x), \lambda) \quad (40)$$

$$RC(f_w(x), \lambda) \leq \frac{\beta^2 + \beta \sqrt{4RC_{\text{emp}}(f_w(x), \lambda) + \beta^2}}{2} + RC_{\text{emp}}(f_w(x), \lambda) \quad (41)$$

由于不等式 (41) 对函数集  $\{C(x, W, \lambda)L(y, f_w(x))\}$  中的所有函数都成立, 所以对于最小化经验风险  $RC_{\text{emp}}(f_w(x), \lambda)$  的函数  $(f_w^a(x), \lambda^a)$  不等式

$$RC(f_w^a(x), \lambda^a) \leq RC_{\text{emp}}(f_w^a(x), \lambda^a) + \frac{\beta^2}{2} + \frac{1}{2} \sqrt{(4RC_{\text{emp}}(f_w^a(x), \lambda^a) + \beta^2)\beta^2} \quad (42)$$

依至少  $1-\mu$  的概率成立, 将式 (37) 代入不等式 (42) 定理得证. 证毕.

由于不等式 (41) 对函数集  $\{C(x, W, \lambda)L(y, f_w(x))\}$  中的所有函数都成立, 所以对于最小化风险  $RC(f_w(x), \lambda)$  的函数  $(f_w^\epsilon(x), \lambda^\epsilon)$  不等式

$$RC(f_w^\epsilon(x), \lambda^\epsilon) \geq RC_{\text{emp}}(f_w^\epsilon(x), \lambda^\epsilon) + \frac{\beta^2}{2} - \frac{1}{2} \sqrt{(4RC_{\text{emp}}(f_w^\epsilon(x), \lambda^\epsilon) + \beta^2)\beta^2} \quad (43)$$

依至少  $1-\mu$  的概率成立. 由定理 3, 不等式 (31), (43) 可推出

$$\begin{aligned} RC(f_w^a(x), \lambda^a) - RC(f_w^\epsilon(x), \lambda^\epsilon) &\leq \\ \frac{1}{2} \sqrt{(4RC_{\text{emp}}(f_w^a(x), \lambda^a) + \beta^2)\beta^2} &+ \\ \frac{1}{2} \sqrt{(4RC_{\text{emp}}(f_w^\epsilon(x), \lambda^\epsilon) + \beta^2)\beta^2} &\quad (44) \end{aligned}$$

依至少  $1-2\mu$  的概率成立.

由定理 2, 3 可知实际风险由经验风险和置信区间两部分组成. 通常学习方法是首先通过选择模型来固定置信区间, 然后通过最小化经验风险泛函来



求最小风险,因为缺乏对置信区间的认识,这种选择往往是依赖于先验知识和经验进行的.为此,Vapnik 提出结构风险最小化原则,即选择最小经验风险与置信区间之和最小的子集,这个子集中使经验风险最小的函数为所求的最优函数.在分布式学习中为了防止被分割的知识丢失,在分布学习阶段没有采用结构风险最小化原则,而在集中学习阶段采用后修剪算法实现结构风险最小化.

2.5 学习算法

学习算法的主体学习过程可描述如下.

初始化学习速度系数  $\mu$ , 相似度门限值  $\vartheta$ ;

- 1. 接收第一个样本向量  $\mathbf{x}$ , 添加第一个神经元  $\mathbf{w}_0$  并置初始值为  $\mathbf{x}$ ;
- 2. 判断学习是否结束: 若否, 则从样本空间中接收一个样本向量, 并计算相似度值  $d_i$ ;
- 3. 由竞争函数判断获胜神经元  $j$ , 若  $d_j > \vartheta$ , 则添加新的神经元并使其突触权值为  $\mathbf{x}$ , 返回步 2, 否则继续;
- 4. 按式(6)更新突触权值, 返回步 2.

由于完整数据集是被随机地分割成不相交的子集, 因此描述某些知识点的样本可能被分割到不同的子集, 为了避免因这类知识点被忽略而降低准确率, 在分布学习阶段各个子神经网络采用不断增长的学习方式, 即当一个样本与当前知识点具有较低的相似度时该样本将成为一个新的知识点并被增加到权重矩阵, 并且该阶段相似度门限值  $\vartheta$  较大. 在分布学习结束后, 各个分割数据的学习结果中包含了一些较完整的知识点和一些没有完全学习到并被分割表示的知识点. 因此, 需要通过集中学习对这些中间结果构成的样本空间进行再学习从而形成完整的知识. 上述过程在解决了样本被分割后一些知识点可能被丢弃的问题的同时也降低了学习结果的泛化能力, 因为样本数据中一些特征有可能被重复表示从而导致过学习. 因此采用后修剪算法来实现结构风险最小化原则. 后修剪算法以训练后的每一个神经元为修剪的候选对象, 将相似的神经元合并为一个神经元或多个(总数比合并前少)神经元. 修剪后新神经元的计算公式为

$$\mathbf{W}_{\text{new}} = (\mathbf{W}_{\text{old1}} \times t_1 + \mathbf{W}_{\text{old2}} \times t_2) / (t_1 + t_2) \quad (45)$$

其中,  $t_1$  为神经元  $\mathbf{W}_{\text{old1}}$  的学习次数,  $t_2$  为神经元  $\mathbf{W}_{\text{old2}}$  的学习次数. 某个神经元学习的次数越多, 其信息在新神经元中占的比例越大.

3 实 验

实验测试首先采用 Circle-in-the-square 基准测

试数据集测试了算法的学习能力, 然后采用 UCI 中的数据集 USCensus1990Data 测试了其大规模数据的学习性能.

3.1 Circle-in-the-square

Circle-in-the-square 是美国国防部高级研究计划署(DARPA)人工神经网络技术(ANNT)计划采用的基准测试问题. 该问题要求神经网络能准确分辨出一单位正方形的点中位于一圆内和圆外的点, 该圆位于正方形中且面积为单位正方形的一半. 文献[38]用 2-n-1BP 神经网络对该基准测试进行了分析. 实验分别测试了当隐层神经元数  $n$  从 5 增加到 100, 权向量个数从 21 增加到 401, 训练集从 150 增加到 14000 时的学习能力, 最后得出当隐层神经元个数为 20~40 个时, 经过 5000 个周期的训练, 神经网络辨别的准确率在 90% 左右. Fuzzy ART-MAP<sup>[28]</sup> 的测试结果表明当  $ART_a$  的神经元数从 12 增加到 121, 训练集从 100 增加到 100000 时其错误率从 11.4% 降低到 2.0%. 测试使用文献[28]的基准测试数据, 该数据集可从 CELEST Technology Website (<http://profusion.bu.edu/techlab/modules>) 下载. 数据集 cis\_train2.txt 中包含 1000 个样本数据, 为了测试分布式学习算法的学习能力, 将数据集按图 2 进行分割, 其中 A1, A2 组成 A 块数据集, B1, B2 组成 B 块数据集. 将上述 A, B, C 块数据分别分发给三个神经网络按分布式学习算法进行学习, 分布学习的结果见图 3~5, 最终的学习结果见图 6.

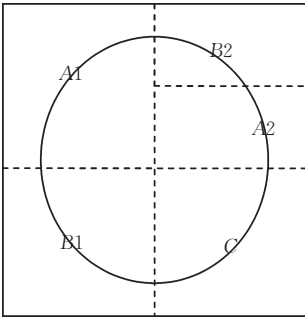


图 2 分割数据的分布情况

采用 SVM 对上述样本进行训练, 核函数采用 ‘rbf’, 交叉验证采用 ‘HoldOut’ 时的训练和识别结果见图 7 和图 8, 其准确率在 94.43%~97.33% 之间. 当相似度门限值 0.05 增加到 0.08 时, 神经网络的训练次数, 神经元数和准确率与其他方法的比较列在表 1 中. 对于分布式学习, 其训练次数为分布学习中训练次数最多的神经网络的训练次数加上集中

学习的训练次数. 在训练的一个周期中,若谐振发生,某个样本可能被多次学习,所以分布式学习的训练次数等价于其它方法的训练周期乘以样本数,从学习结果可以看出分布学习并没有丢弃被分割的知识,学习结果能较好地分辨出点所在的区域.

表 1 Circle-in-the-square 测试结果

	训练次数	正确率/%	神经元数
分布神经网络	518~621	96~98	91~196
ARTMAP	(1×100)~ (1×100000)	88.6~98	12~121
BP	(5000×150)~ (5000×14000)	90	21~401

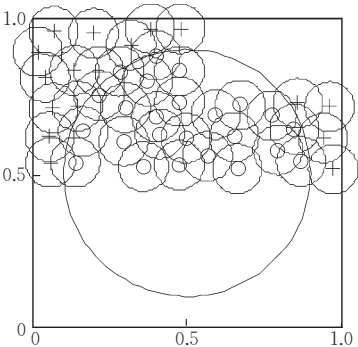


图 3 A 部分数据的分布学习结果

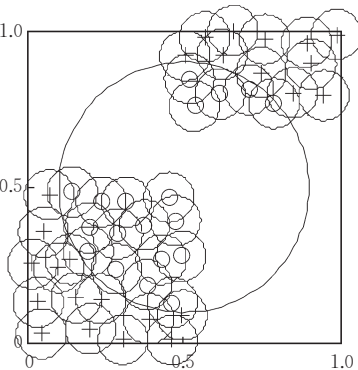


图 4 B 部分数据的分布学习结果

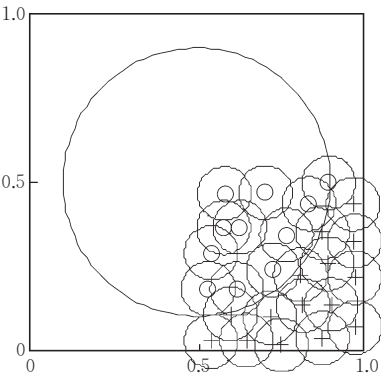


图 5 C 部分数据的分布学习结果

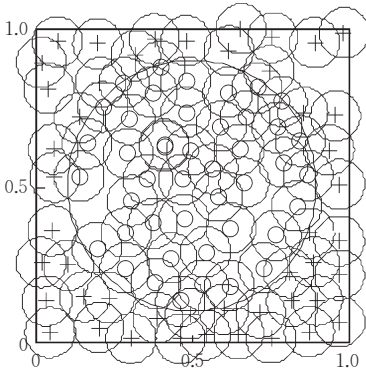


图 6 集中学习结果

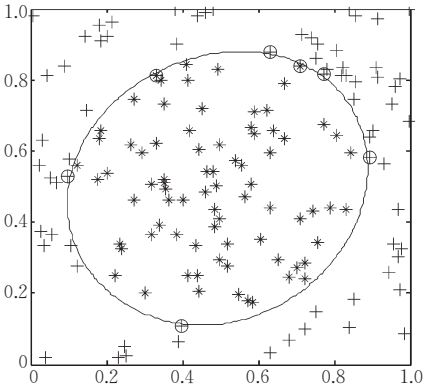


图 7 SVM 学习结果

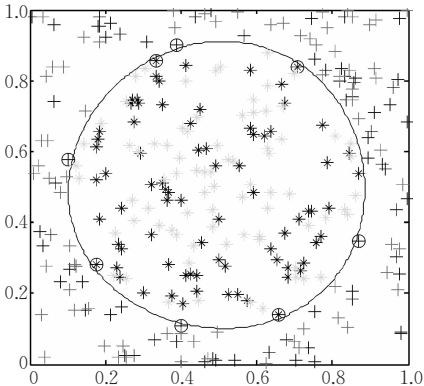


图 8 SVM 识别结果

3.2 USCensus1990Data

上面实验结果表明,即使样本中的知识点包含在分割的子样本集中,分布神经网络算法也能有效地学习到并与单个神经网络学习具有等价的性能.对于大数据集的测试,本文采用了 UCI 中的 US-Census1990Data,这个 352Megabytes 数据集包含了由 68 个属性组成的 2458285 条记录.实验中将数据集按记录顺序分割成 12 个不相交子集,前 11 个子集每个包含 200000 条记录,最后一个子集包含最后的 258285 条记录.实验测试了在分布学习下各个子神经网络的神经元个数和集中学习后神经元个数



及总共消耗的时间,并将其与采用一个神经网络对全部样本进行学习的结果进行了比较。

实验中分布学习时的门限值为 15,集中学习时的门限值为 10,单个神经网络学习的门限值为 18. 学习后的神经元个数和消耗时间情况见表 2. 其中分布学习阶段前 11 个子神经网络消耗时间在 12 分钟左右,第 12 个子神经网络耗时较多是因为其训练样本比其它子集多了 58285 条. 由上述结果可知分布神经网络消耗的总时间为  $16.8+0.5=17.3\text{min}$ ,而单个神经网络消耗的时间是其 6 倍多.

表 2 USCensus1990Data 测试结果

	消耗时间 /min	神经元数 /个		消耗时间 /min	神经元数 /个
分布 1	12.3	217	分布 8	13.4	242
分布 2	12.2	221	分布 9	11.8	210
分布 3	12.3	212	分布 10	11.9	213
分布 4	12.4	215	分布 11	12.7	229
分布 5	12.8	231	分布 12	16.8	232
分布 6	11.9	208	集中	0.5	125
分布 7	12.1	215	单个	105.1	147

4 结 论

为了对大规模数据进行归纳学习,KDD 研究人员提出了对数据进行采样学习,将数据分割后分布\并行学习等 Scaling up 学习方法. Scaling up 学习在面临着学习算法过学习难题的同时更面临着因数据量巨大导致的欠学习难题. 数据分割后对数据进行分布\并行处理面临的主要问题是如何将各个子数据集的学习结果进行合并,从而使分散的知识组合成最终的知识模型. 本文提出了基于 Hebb 规则的分布神经网络学习方法,Hebb 规则的局部学习特征使被分割到各个子集的部分知识能够在分布学习阶段得到保留并在集中学习阶段被提取出来. 对 Circle-in-the-square 的实验证明了该分布神经网络的准确性与单个神经网络相当. 通过 USCensus1990Data 实验表明该学习方法通过分布学习不仅解决了大规模数据样本学习时的空间约束问题,如即使在 1GB 的内存容量下 Matlab 都无法装载全部的 USCensus1990Data 样本数据,而且分布处理大大提高了系统的整体学习速度.

参 考 文 献

[1] Wei C P, Lee Y H, Hsu C M. Empirical comparison of fast partitioning-based clustering algorithms for large data sets. Expert Systems with Applications, 2003, 24(4): 351-363

[2] Peter W, Chiochetti J, Giardina C. New unsupervised clustering algorithm for large datasets//Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington D. C. , 2003: 643-648

[3] Gursoy A. Data decomposition for parallel  $k$ -means clustering//Proceedings of the 5th International Conference on Parallel Processing and Applied Mathematics. Czestochowa, Poland, 2003: 241-248

[4] Ceglar A, Roddick J F. Association mining. ACM Computing Surveys, 2006, 38(2): 1-42

[5] Parthasarathy S, Zaki M J, Ogihara M, Li W. Parallel data mining for association rules on shared-memory systems. Knowledge and Information Systems, 2001, 3(1): 1-29

[6] Jia C Y, Gao X P. Multi-scaling sampling: An adaptive sampling method for discovering approximate association rules. Journal of Computer Science and Technology, 2005, 20(3): 309-318

[7] Tuv E, Borisov A, Torkkola K. Best subset feature selection for massive mixed-type problems//Proceedings of the 7th International Conference on Intelligent Data Engineering and Automated Learning. Burgos, Spain, 2006: 1048-1056

[8] Tang W Y, Mao K Z. Feature selection algorithm for data with both nominal and continuous features//Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining. Hanoi, Vietnam, 2005: 683-688

[9] John G, Langley P. Static versus dynamic sampling for data mining//Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining. Portland, Oregon, 1996: 367-370

[10] Provost F, Jensen D, Oates T. Efficient progressive sampling//Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego, California, United States, 1999: 23-32

[11] Meek C, Thieson B, Heckerman D. The learning-curve sampling method applied to model-based clustering. Journal of Machine Learning Research, 2002, 2: 397-418

[12] Jia C Y, Gao X P. Multi-scaling sampling: an adaptive sampling method for discovering approximate association rules. Journal of Computer Science and Technology, 2005, 20(3): 309-318

[13] Wu X, Lo H W. Multi-layer incremental induction//Proceedings of the 5th Pacific Rim International Conference on Artificial Intelligence. Singapore, 1998: 24-32

[14] Fürnkranz J. Integrative windowing. Journal of Artificial Intelligence Research, 1998, 8: 129-164

[15] Prodromidis A L, Stolfo S J. A comparative evaluation of meta-learning strategies over large and distributed data sets//Proceedings of the 16th International Conference on Machine Learning. Bled, Slovenia, 1999: 18-27

[16] Giraud-Carrier C, Vilalta R, Brazdil P. Introduction to the special issue on meta-learning. Machine Learning, 2004, 54(3): 187-193

[17] Littlestone N, Warmuth M K. The weighted majority algorithm. Information and Computation, 1994, 108(2): 212-261

[18] Chan P K, Stolfo S J. On the accuracy of meta-learning for scalable data mining. Journal of Intelligent Information Systems, 1997, 8(1): 5-28

- [19] Merz C J. Using correspondence analysis to combine classifiers. *Machine Learning*, 1999, 36(1~2): 33-58
- [20] Wolpert D H. Stacked generalization. *Neural Networks*, 1992, 5(2): 241-259
- [21] Wanas N, Kamel M S, Auda G, Karray F. Feature-based decision aggregation in modular neural network classifiers. *Pattern Recognition Letters*, 1999, 20(11): 1353-1359
- [22] Raudys S. Trainable fusion rules. I. Large sample size case. *Neural Networks*, 2006, 19(10): 1506-1516
- [23] Nguyen M H, Abbass H A, McKay R I. A novel mixture of experts model based on cooperative coevolution. *Neurocomputing*, 2006, 70(1~3): 155-163
- [24] Chan P K, Fan W, Prodromidis A L, Stolfo S J. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems*, 1999, 14(6): 67-74
- [25] Hall L O, Chawla N, Bowyer K W. Decision tree learning on very large data sets//*Proceedings of the IEEE SMC Conference*. San Diego, California, 1998: 2579-2584
- [26] Amado N, Gama J, Silva F. Parallel implementation of decision tree learning algorithms//*Proceedings of the 10th Portuguese Conference on Artificial Intelligence*. Porto, Portugal, 2001: 6-13
- [27] Todorovski L, Dzeroski S. Combining classifiers with meta decision trees. *Machine Learning*, 2003, 50(3): 223-249
- [28] Carpenter G A, Grossberg S, Markuzon N, Reynolds J H, Rosen D B. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, 1992, 3(5): 698-713
- [29] Howley T, Madden M G, O'Connell M L, Ryder A G. The effect of principal component analysis on machine learning accuracy with high-dimensional spectral data. *Knowledge-Based Systems*, 2006, 19(5): 363-370
- [30] Mingoti S A, Lima J O. Comparing SOM neural network with Fuzzy c-means, K-means and traditional hierarchical clustering algorithms. *European Journal of Operational Research*, 2006, 174(3): 1742-1759
- [31] Nair T M, Zheng C L, Fink J L. Rival penalized competitive learning (RPCL): A topology-determining algorithm for analyzing gene expression data. *Computational Biology and Chemistry*, 2003, 27(6): 565-574
- [32] Tian D X, Liu Y H, Wei D. A dynamic growing neural network for supervised or unsupervised learning//*Proceedings of the 6th World Congress on Intelligent Control and Automation*. Dalian, China, 2006: 2886-2890
- [33] Andrew L. Analyses on the generalized lotto-type competitive learning//*Proceedings of the 2nd International Conference on Intelligent Data Engineering and Automated Learning*. Hong Kong, 2000: 9-16
- [34] Geman S, Bienenstock E, Doursat R. Neural networks and the bias/variance dilemma. *Neural Computation*, 1992, 4(1): 1-58
- [35] Bishop C M. Training with noise is equivalent to Tikhonov regularization. *Neural computation*, 1995, 7(11): 108-116
- [36] Bottou L, Vapnik V N. Local learning algorithms. *Neural Computation*, 1992, 4(6): 888-900
- [37] Vapnik V N, Bottou L. Local algorithms for pattern recognition and dependencies estimation. *Neural Computation*, 1993, 5(6): 893-909
- [38] Kevin J L, Michael J W. Learning to tell two spirals apart//*Proceedings of the 1988 Connectionist Models Summer School*. 1988: 52-61



**TIAN Da-Xin**, born in 1980, Ph. D. candidate. His main research interests include network security and protocol, machine learning and artificial neural network.

Ph. D. supervisor. His main research interests include computer communication and network, mobile IP, and QoS.

**LI Bin**, born in 1960, associate professor. Her main research interests include artificial intelligence and data mining.

**WU Jing**, born in 1973, Ph. D. candidate. Her main research interests include network security and machine learning.

**LIU Yan-Heng**, born in 1958, Ph. D., professor and

## Background

The knowledge discovery and data mining community has challenged itself to develop inductive learning algorithms that scale up to large data sets. Many diverse techniques have been proposed and implemented for scaling up inductive algorithms. The three main approaches are: design a fast algorithm, partition the data and use a relational representation. Two important characters of neural network are: distributed, knowledge representation is distributed across many processing units; parallel, computations take place in parallel across these distributed representations. Although its knowledge representation is distributed, its learning algorithm is con-

centrated, since it requires all the training data to be submitted to the network one by one until the network is stable after one or more epochs. Thus for many realistic problems and databases such as astronomy data, biomedical data, bioinformatics data, etc. it is clearly untenable. This paper presents a distributed neural network learning algorithm. This research is supported by the National Natural Science Foundation of China under grant No. 60573128 and the National Research Foundation for the Doctoral Program of Higher Education of China No. 20060183043.