基于交叉分组技术的集成算法研究*)

朱小飞[1,2] 陈 龙[1] 王国胤[1]

(重庆邮电大学计算机科学与技术研究所 重庆 400065)[1] (重庆工学院数理学院 重庆 400050)[2]

摘 要 集成学习主要通过扰动训练数据集来产生较强泛化能力。研究者们提出了各种各样的方法来实现这一目标,但如何扰动训练数据集以达到最佳的泛化能力并没有被深入研究。本文中,提出了对训练数据集进行扰动的交叉分组(cross-grouping)方法,通过改变交叉因子以实现对训练数据集不同程度的扰动,从而实现当集成规模较小时,得到更强的泛化能力。实验表明,当选择合适的交叉因子时,CG-Bagging 泛化能力要强于 Bagging 和 Boosting,略优于 Decorate 和 Random Forests。

关键词 机器学习,集成学习,泛化能力,交叉分组

Cross-Grouping Based Ensemble Learning

ZHU Xiao-Fei[1,2] CHEN Long[1] WANG Guo-Yin[1]

(Institute of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065)[1]

(School of Mathematics Sciences, Chongqing Institute of Technology, Chongqing 400050)[2]

Abstract Ensemble learning is through disturbing training data to generate strong generalization ability. Researchers have proposed a variety of methods to achieve this goal, but how to achieve the best generalization ability by disturbing training data has not been thorough study. In this paper, we give a novel method called cross-grouping to disturb training data, and achieve different degrees of the disturbance of training data by varying cross-factor. Based on this method, we can achieve stronger generalization ability when the ensemble size is small. Experiment shows that with an appropriate choice of cross-factor, the generalization ability of CG-Bagging is stronger than that of Bagging and Boosting, and slightly better than that of Decorate and Random Forests.

Keywords Machine learning, Ensemble learning, Generalization ability, Cross-grouping

1 引言

集成学习(ensemble learning)技术利用基学习器生成多个分类模型,然后综合其输出结果,可以显著地提高学习系统的泛化能力,是目前机器学习最重要的研究方向之一[1]。Krogh 和 Vedelsby[2]给出了著名的衡量学习系统泛化能力的等式 $E=\bar{E}-\bar{D}$,其中,$\bar{E}$ 为集成中分类模型的平均泛化误差,$\bar{D}$ 为集成中各分类模型之间的平均差异性。该等式指出学习系统的泛化误差等于集成中各分类模型的平均泛化误差和各分类模型的平均差异性之差。因此,要增强学习系统的泛化能力,一方面应尽可能提高各分类模型的平均泛化能力,另一方面应尽可能地增大各分类模型之间的平均差异。现在已经有很多集成学习算法,比较著名的算法有 Bagging 算法[3]和 Boosting 算法[4]。

Bagging 在训练阶段,各学习器的训练数据集由原始训练数据集利用可重复取样(bootstrap sampling)技术获得,训练数据集的规模通常与原始训练数据集相当。这样,原始训练数据集中某些实例可能在新的训练数据集中出现多次,而另外一些实例则可能一次也不出现。为了得到较好的分类结果,Bagging 需要使用大规模的集成来解决问题。在 Boosting 中,首先对原始训练数据集中所有实例分配相同的权重,使用基学习器得到一个新的分类模型;然后根据该分类模型的输出调整各实例的权重:增加被该分类模型误分类的实例的权重,减小被其正确分类的实例的权重,在此基础上得到新的分类模型,依次类推,得到若干个能力互补的分类模型。总的来说,Boosting 比 Bagging 具有更好的分类性能,但是在少数情况下,Boosting 会由于过度拟合数据产生效果非常差的分类模型。

在 Bagging 和 Boosting 的基础之上,研究人员还提出了一些改进的方法,例如 P. Melville 和 R. J. Mooney 等人[5]提出了一种利用人工生成的实例来增加各分类模型之间差异性的算法 Decorate,从而实现当数据集有限时可以产生差异性更大的分类模型。Breiman 等人[6]提出随机森林(Random Forests)算法,其通过同时采用两种途径来增强各分类模型差异性:(1) 类似于 Bagging,采用可放回抽样(sample with replacement)产生新数据集;(2) 在决策树的每一个节点上,在随机选择部分属性中寻找最优划分属性;周志华等人[7]提出的选择性集成 (selective ensemble) 算法,认为通过选择部分分类模型来构建集成要优于使用所有分类模型构建的集成。

以上的集成学习方法致力于通过各种途径来扰动训练数据集。但是,如何更好地实现对训练数据集的扰动以达到最佳的泛化能力并没有进行深入研究。本文针对此问题进行研究,提出了 CG-Bagging 算法,并将其与目前几种著名的算法

*)资助项目:国家自然科学基金项目(No. 60373111);重庆市教委科学技术研究项目(No. KJ060517);重庆市自然科学基金重点资助项目(2005BA2003);重庆市优秀中青年骨干教师资助计划。朱小飞 硕士研究生,主要研究方向为机器学习和智能信息处理;陈 龙 副教授,主要研究方向为网络安全和智能信息处理;王国胤 教授,博士生导师,主要研究方向为粗集理论,粒度计算,神经网络,智能信息系统,网络安全,多媒体数据处理。

作了比较，实验表明该算法能够在集成规模较小时产生更强的泛化能力。

2 交叉分组(Cross-Grouping)方法

在集成学习中，个体分类模型的分类精度与其对应的训练数据集有密切的关系，一般情况下，训练数据集在原始训练数据集中所占的比例越大，则个体分类模型的泛化能力越强。同时由于各训练数据集之间的相似实例数也越来越多，从而导致各分类模型的之间差异性越小。因此，在对训练数据集进行扰动时，应同时考虑各分类模型的泛化能力与各分类模型之间差异性这两个相互制约的因素。

首先，为了便于研究，我们作了如下限定：

(1) 只考虑原始训练数据集中的实例是否在训练数据集中出现，而忽略其出现频率。举例来说，如果集成中共有 3 个训练数据集：A、B 和 C，实例 x 在训练数据集 A 中出现 3 次，在训练数据集 B 中出现 2 次，而在训练数据集 C 中没有出现，则实例 x 在集成中出现的次数记为 2 次(训练数据集 A 和训练数据集 B)，而不是 5 次。做这样的限定是合理的，因为从经验的角度来看，一个实例在训练数据集中是否出现对分类模型所产生的影响要远大于其出现次数对分类模型所产生的影响。

(2) 原始训练数据集中的所有实例在集成中出现的次数相同，且各训练数据集所包含的训练实例数相同(不考虑重复抽取的实例)。该限定的目的是为了使原始训练数据集中的每个实例在集成中被均等地对待，使得每个实例对集成泛化能力的贡献相同。之所以做这样的限定是因为我们没有先验知识说明某些实例对集成泛化能力的贡献要明显优于其他的实例对集成泛化能力的贡献，因此限定每个实例对集成泛化能力的贡献相同是合理的。

在上述限定条件下，我们分析在 Bagging 和 Boosting (Weka[8] 中实现的是 AdaBoostingM1[9])集成方法中，原始训练数据集中各实例在集成中出现的次数分布情况。

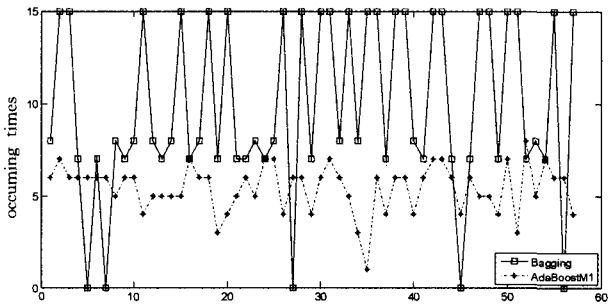

图 1 Bagging 和 AdaBoostM1 中各实例出现次数

图 1 中给出了采用 UCI 机器学习数据库[10] 中的 labor 数据集进行实验的结果，从图中可以看到，采用 Bagging 集成方法时，有 8.8% (5/57)的实例在集成中的出现次数为 0，这意味着这些样本对各分类模型的泛化能力与各分类模型之间差异性没有贡献；40.4%(23/57)的实例在集成中的出现次数为 15 次(默认的集成规模为 15)，即这些实例在所有训练数据集中均出现，因而这些实例对各分类模型之间差异性没有贡献，但对各分类模型的泛化能力的贡献达到最大。而在 Boosting 集成方法中，不存在实例在集成中的出现次数为 0 或 15，这说明在采用 Boosting 集成方法时，原始训练数据集中每个实例在各分类模型的泛化能力与各分类模型之间差异性之间作了折中，因而增强了集成的泛化能力。此外，Bagging 和 Boosting 集成方法共同的特点在于每个实例在集成中出现的次数变动较大(从图中可以看出，Bagging 的变动幅度要大于 Boosting 的变动幅度)。从上述分析来看，对于原始训练数据集中每个实例，如何选择一个合适的出现次数使得在各分类模型的泛化能力与各分类模型之间差异性得到最佳的平衡，是达到最优的泛化能力的重要途径。

2.1 交叉因子 r 及其选择

根据假设条件(2)，我们约定各实例出现的次数相同，这样将产生一个变动幅度为 0 的次数分布曲线。因此，要分别找到为各实例合适的出现次数的问题就简化为只要找到一个一致的合适出现次数的问题。

定理 1 令 K 表示集成规模(即集成中分类模型的数目)，M 表示原始数据集的规模，L 表示新数据集的规模，N 表示原始数据集中的各实例在集成中的出现次数，则有：

$$K*L=M*N \tag{1}$$

证明：在集成中有 K 个分类模型，则其需要 K 个新数据集来产生这 K 个分类模型，因此集成中真正包含的训练实例数为这 K 个新数据集所包含的所有训练实例数，即为 $K*L$；此外，原始数据集中共含有 M 个训练实例，又每个实例在集成中出现次数为 N，因此 $M*L$ 即为集成中包含的所有训练实例数，与等式左边含义相同，问题得证。

集成规模 K 为事先指定的一个很小的常数，原始数据集的规模 M 是依赖于具体的研究领域事先确定的，因此原始数据集中的各实例在集成中的出现次数 N 就转化为关于新数据集的规模 L 的函数关系式：

$$N=K*L/M \tag{2}$$

令交叉因子 r 表示各新数据集包含训练实例在原始训练数据集中所占的比例(不考虑从原始训练数据集中重复抽取的样本)，即 $r=L/M$，则：

$$N=r*K \tag{3}$$

因为新数据集是原始数据集抽取样本产生的，忽略重复采样得到的样本，可看作是原始数据集的子集，即 $L<M$，所以 r 为一个介于 0 和 1 之间的数。通过改变 r 的大小，可得到不同性能的集成差异性。

2.2 Cross-grouping

为了确定合适的 r 值，我们提出了交叉分组(Cross-grouping)技术，其基本思想如下：给定原始训练数据集 S 和 K 个空的子数据集 $T_i(i=1,\cdots,K)$，然后将 S 中的第 1 个实例放入到 $T_1,\cdots,T_{r*K}$ 中，第 2 个实例放入到 $T_2,\cdots,T_{r*K+1}$ 中，第 i 个实例放入 $T_{(i-1) \bmod K+1},\cdots,T_{(i+r*K-2) \bmod K+1}$ 中，依次类推，直到 S 中每一个实例都出现在新的子数据集中。交叉分组的算法伪码描述如下：

Input:
M—number of given training instances
K—number of ensemble size
S—original training set: $\langle(x_1,y_1),\cdots,(x_m,y_m)\rangle$ >with labels $y_i\in Y$
r—proportion of sub-training set in original traninng set
Output: T—a bag of sub-training sets $\{T_1,T_2,\cdots,T_K\}$
Initialize: set a roulette with K slots,
1. Fof each iteration $i=1:M$,
2. {
3. move instance (x_i,y_i) to these slots from $T_{(i-1) \bmod K+1}$ to $T_{(i+r*K-2) \bmod K+1}$;
4. }
5. T is a gab of the sub-training sets.

我们选取 UCI 机器学习数据库中的四个数据集 *autos*, *iris*, *glass*, *lymph* 来进行实验，结果如图 2 所示。在所有数

据集上，集成的泛化能力先随着 r 的增加不断增加，当 r 值介于 0.6～0.8 之间时达到最大，然后呈下降趋势。在 CG-Bagging 算法中，我们将选择 0.7 作为 r 的默认值。

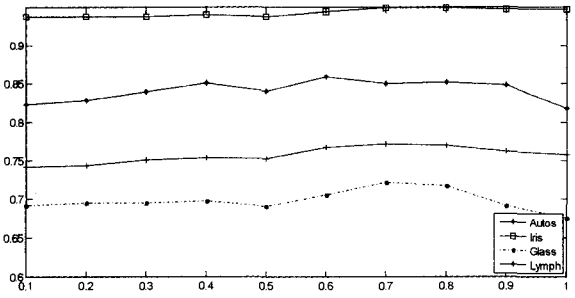

图 2　泛化能力随交叉因子 r 的变化曲线

3　CG-Bagging 算法

首先采用交叉分组(cross-group)的方法生成 K 个规模为 $r*M$ 的新训练数据集。然后，在新生成的子训练数据集上采用基学习器学习得到 K 个学习模型，通过投票法进行集成，CG-Bagging 算法伪码描述如下：

Input:
　　BaseLearn—base learning algorithm
　　Cross-group—the Cross-Group Algorithm
　　K—number of ensemble size
　　S—the original training set: $\langle(x_1,y_1),\cdots,\langle x_m,y_m\rangle\rangle$ with labels $y_i\in Y$
　　T—the derived training set: $\{T_1,T_2,\cdots,T_K\}$
Output: $C^*(x)$-the class with the most votes
1. T=Cross-group(S)
2. For each iteration $i=1\cdots K$
3. {
4.　Train base learner, C_i=BaseLearn(T_i)
5. }
6. The final classifier $C^*(x)$ is formed by aggregating the K classifiers
7. To classify an instance x, a vote for class y is recorded by every classifier $C_i(x)=y$
8. $C^*(x)$ is the class with the most votes.

4　实验测试

本文将 CG-Bagging 算法与 Bagging、Boosting(这里使用 Adaboosting 算法)、Decorate 以及 RandomForest 算法进行了比较。实验采用的 UCI 机器学习数据库[10]中的 15 个数据集(见表 1)对算法进行了实验测试，同时使用 10 倍交叉验证(10-fold cross validation)方法。CG-Bagging 使用 C4.5 算法[11,12](在 weka 中称为 J48 算法)作为基学习器，集成规模默认为 15，其他算法的参数均采用其在 Weka 中的默认值。此外，我们使用了双边配对 t 检验，其中 win(v)，loss(*)和 draw 分别表示在显著性水平 0.05 下，CG-Bagging 算法与其余四个算法相比，其分类准确率明显优于、明显劣于以及基本相当的数据集个数。

表 2 列出了每种算法的错误率以及其标准差。从表 2 中可知，CG-Bagging 算法在数据集 audio，autos，segment，soybean 和 vote 上要明显优于 Bagging 算法，在其他数据集上，其与 Bagging 算法基本相当；在数据集 heart-c 上，CG-Bagging 算法明显劣于 AdaBoostM1 算法，但在数据集 anneal，audio，autos，glass，segment 和 soybean 上，CG-Bagging 算法明显优于 AdaBoostM1 算法，尤其是在数据集 segment，soybean 上，AdaBoostM1 算法的错误率分别达到 71.5%和 72%，而 CG-Bagging 算法分别只有 2.5%和 7.4%，产生这种现象的原因主要是因为使用 AdaBoostM1 算法有时会过度拟合(over-fitting)训练数据集，而使用 CG-Bagging 算法可以避免产生过度拟合现象。CG-Bagging 算法分别在数据集 vote 和 audio 明显优于 Decorate 算法和 RandomForest 算法。

表 1　实验数据有关信息

Names	Cases	Classes	Attributes	
			Numeric	Nominal
anneal	898	6	9	29
audio	226	6	—	69
autos	205	6	15	10
breast-w	699	2	9	—
colic	368	2	10	12
credit-a	690	2	6	9
glass	214	6	9	—
heart-c	303	2	8	5
hepatitis	155	2	6	13
iris	150	3	4	—
labor	57	2	8	8
lymph	148	4	—	18
segment	2310	7	19	—
soybean	683	19	—	35
vote	435	2	—	16

总的来看，CG-Bagging 算法要优于 Bagging 算法和 AdaBoostM1 算法，略胜于 Decorate 算法和 RandomForest 算法。

表 2　实验结果

Dataset	CG-Bagging	Bagging	AdaBoostM1	Decorate	RandomForest
anneal	0.011±0.010	0.012±0.012	0.164±0.005*	0.013±0.010	0.006±0.008
audio	0.174±0.078	0.240±0.070*	0.535±0.022*	0.191±0.076	0.235±0.080*
autos	0.149±0.085	0.335±0.097*	0.551±0.062*	0.169±0.084	0.183±0.081
breast-w	0.046±0.028	0.045±0.025	0.049±0.025	0.042±0.024	0.042±0.021
colic	0.156±0.056	0.152±0.056	0.175±0.054	0.157±0.057	0.152±0.053
credit-a	0.147±0.039	0.144±0.040	0.152±0.041	0.145±0.041	0.149±0.041
glass	0.279±0.085	0.276±0.088	0.551±0.031*	0.287±0.096	0.239±0.087
heart-c	0.227±0.065	0.198±0.070	0.165±0.072 v	0.221±0.068	0.197±0.065
hepatitis	0.201±0.098	0.181±0.070	0.186±0.081	0.185±0.103	0.169±0.080
iris	0.051±0.050	0.058±0.056	0.046±0.057	0.049±0.051	0.058±0.052
labor	0.175±0.154	0.164±0.170	0.116±0.139	0.116±0.119	0.131±0.135
lymph	0.227±0.093	0.226±0.096	0.246±0.107	0.218±0.114	0.194±0.090
segment	0.025±0.011	0.034±0.012*	0.715±0.002*	0.022±0.009	0.023±0.011
soybean	0.074±0.029	0.130±0.037*	0.720±0.021*	0.062±0.028	0.081±0.029
vote	0.030±0.024	0.043±0.027*	0.046±0.031	0.055±0.033*	0.041±0.027
Average	0.131	0.149	0.295	0.129	0.127
[v/ /*]		[5/10/0]	[6/8/1]	[1/14/0]	[1/14/0]

(下转第 193 页)

5. 重复 1～3 步骤，直到获得新的 $SV5$，$SV6$；

6. 将通过反馈得到的新的 $SV5$，$SV6$ 合并，得到最后的输出 $SV7$ 和 SVM 决策函数；

7. 如果 $SV5$($SV6$)与通过反馈得到的 $SV5$($SV6$)相同，或者其差集中元素固定，则算法结束；

8. 如果不满足 7 中的条件，则返回第 5 步，重新开始训练。

5 实验及结果分析

先将现实中需要分类的数据集平均分成 4 个训练子集，分别训练得到 4 个 SVM 分类器，用事先人工分类的 4235 个数据样本作为测试集，生成的 78963 个样本为数据集，其维数为 26。

所有的程序使用 MATLAB7.0 编写，通过使用 MATLAB SVM Toolbox 训练仿真比较。采用标准的 SVM 算法以及径向基函数，即

$$K(x,y)=\exp\left[-\frac{(x-y)^2}{2\delta^2}\right]$$

为了说明分级并行算法的性能，我们在同一数据集上进行了三次实验。

实验 1 为标准 SVM，实验 2 为分级并行 SVM(无反馈)，实验 3 为分级并行 SVM(有反馈)。实验 1 和实验 2 使用传统的方法选择支持向量，实验 3 中使用本文改进的分级方法。表 1 列出了这三个实验的比较结果，最终输出的是支持向量的个数和分类精度。

表 1 实验结果

	E1	E2	E3
Number of SVs	2025	2356	1935
Output precision(%)	94.46	92.07	96.18
Training time(s)	30	18	25.2
feedback	no	no	yes

实验结果分析：实验 3 中，训练时间要比标准 SVM 算法少，具有较高的分类精度。实验 2 中，具有最少训练时间，因其忽略了所有的反馈，但分类精度最低，这个结果符合文中的分析与预测。实验 1 中，无论是分类精度，还是训练时间都无法达到让人满意的程度。

结论和进一步研究 本文将信息融合技术与 Mobile Agent 相结合，改进了 OODA 模型，提出了一种基于 SVM 的分级信息融合算法。该算法以分级结构为基础，分别由多个 SVM 分类器进行并行训练。在信息融合过程中，采用支持向量机的分级学习算法是可行的，可有效解决小样本、非线性参数之间存在模糊关系的信息融合问题。反馈和支持向量的选择是算法对标准反馈分级结构的改进，由于各局部优化独立解决，整体的存储空间和计算时间的消耗被大大降低。实验结果表明，本文中的训练算法可达到更为满意的分类效果，并可以得到较高的分类精度。此外，核函数的类型及相关参数的选择对融合精度有一定影响，如何优化，我们将对这些问题进行进一步的研究。

参 考 文 献

1 Varshney P K. Distributed Detection and Data Fusion. New York: Springer- Verlag, 1996

2 Shahbazian E, Dale E, Blodgett P L. The Extended OODA Model for Data Fusion Systems. In: Proceeding of International Conference on Information Fusion, USA, 2001

3 Shaban K B. Information Fusion in a Cooperative Multi-agent System for Web Information Retrieval. In: Proceedings of the Fifth International Conference on Information Fusion, Singapore, 2002

4 Kim Yongseog. Information Fusion via a Hierarchical Neural Network model. Journal of Computer Information Systems, 2005(4): 1～13

5 田盛丰，黄厚宽. 基于支持向量机的数据库学习算法. 计算机研究与发展，2000，37(1)：7～22

6 Vapnik V N. The Nature of Statistical Learing Theory. New York: Springer- Verlag, 1995

7 Kumar R, Wolenetz M, Agarwalla B. A Framework for Distributed Data Fusion. Information Fusion, 2007, 8(3): 227～251

8 Andler S F, Niklasson L, Persson O B. A Information Fusion from Databases, Sensors and Simulations: a Collaborative Research Program. In: 29th Annual IEEE/NASA Software Engineering Workshop, 2006

9 Wen Y M, Lu B L. A cascade method for reducing training time and the number of support vectors. In: Proceeding of International Symposium on Neural Network, Dalian, 2004

(上接第 160 页)

结论 本文针对 Bagging 和 Boosting 的不足，提出了一种新的算法 CG-Bagging。该算法能够在小规模集成的基础上得到较好的泛化能力，并且避免类似于 Boosting 产生的过度拟合的不足。在 UCI 机器学习数据库上的实验结果表明，CG-Bagging 的泛化能力略强于 Decorate 和 RandomForest，但其效率远优于 Bagging 和 Boosting。进一步的工作包括如何从其他途径生成平均泛化误差小且个体差异度大的分类模型，以提高集成的泛化能力。

参 考 文 献

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

2 Krogh A, Vedelsby J. Neural network ensembles, cross validation and active learning. In: G Tesauro, D S Touretzky and T K Leen, eds. Advances in Neural Information Processing Systems 7. Cambridge, MA: MIT Press, 1995

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

4 Schapire R E. The boosting approach to machine learning: An overview. In: MSRI Workshop on Nonlinear Estimation and Classification, 2001

5 Melville P, Mooney RJ. Constructing Diverse Classifier Ensembles using Artificial Training Examples. In: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence. Acapulco, Mexico, 2003. 505～510

6 Breiman L. Random Forests. Machine Learning, 2001, 45(1): 5～32

7 Zhou Z H, Wu J, Tang W. Ensembling Neural Networks: Many Could Be Better Than All Artificial Intelligence, 2002, 137: 239～263

8 Witten IH, Frank E. Data Mining: Practical Machine Learning Tools and Techniques, 2nd edition. Morgan Kaufmann, San Francisco, 2005

9 Freund Y, Schapire R E. Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth International Conference on Machine Learning. San Francisco, 1996. 148～156

10 Blake C L, Merz C J. UCI Repository of Machine Learning Databases. Available at: http://www.ics.uci.edu/～mlearn/Mlrepository.html, 1998

11 Quinlan J R. Bagging, Boosting, and C4.5. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence. Cambridge, MA: AAAI Press/MIT Press, 1996

12 Quinlan J R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993