Hybrid Adaboost based on Genetic Algorithm for Gene Expression Data Classification

Yaqiong Meng
China Jiliang University
Hangzhou, China
825726214@qq.com

Huijuan Lu*
China Jiliang University
Hangzhou, China
hjlu@cjlu.edu.cn

Ke Yan
China Jiliang University
Hangzhou, China
yanke@qq.com

Minchao Ye
China Jiliang University
Hangzhou, China
yeminchao@cjlu.edu.cn

ABSTRACT

This paper presents a hybrid Adaboost algorithm. The decision groups are chosen as weak classifiers, which consist of K nearest neighbor algorithm, Naïve Bayes and decision tree. When the weak classifiers are promoted to strong classifier, the genetic algorithm is used to optimize the discourse right of each weak classifier. Experiments show proposed algorithm compared with the weak algorithm integration algorithm with only a single algorithm, the proposed algorithm is superior.

CCS CONCEPTS

· **Theory of computation** → Theory and algorithms for application domains→Machine learning theory→Boosting

KEYWORDS

Adaboost algorithm, genetic algorithm, K nearest neighbor algorithm, Naïve Bayes, decision tree

ACM Reference format:

Yaqiong Meng, Huijuan Lu, Ke Yan and Minchao Ye. 2017. Hybrid Adaboost based on Genetic Algorithm for Gene Expression Data Classification. In *Proceedings of ChineseCSCW' 17, Chongqing, China, September 22-23, 2017,* 2 pages.

https://doi.org/10.1145/3127404.3127466

1 引言

本文以癌症基因数据为背景[1]。提出一种基于遗传算法的混合 Adaboost 算法对基因数据进行分类，它的弱分类器是有 KNN[2]，NB[3]，DT[4]组成的一个决策小组，最后利用遗传算法优化各弱分类器的话语权。本文算法的分类效果要优于：Bagging、随机森林（RF）、旋转森林（RoF）等集成分类算法以及非集成算法 ELM。

* Corresponding author.

ChineseCSCW '17, September 22–23, 2017, Chongqing, China

ACM ISBN 978-1-4503-5352-6/17/09.
https://doi.org/10.1145/3127404.3127466

2 决策小组

决策小组为 NB，DT，KNN 的随机组合，每个决策小组有三个算法组成，样本的类别由 KNN，NB 和 DT 三种算法通过投票法决定。如图 1 所示。本文将决策小组作为 Adaboost 算法[5]的弱分类器。

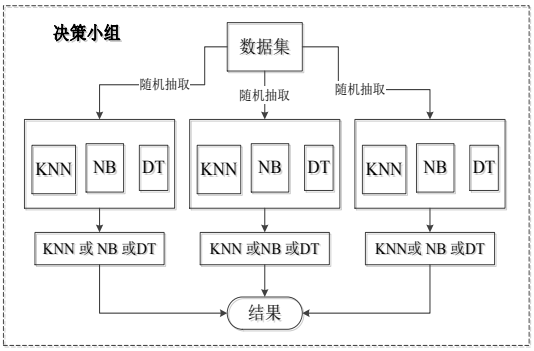

图 1 决策小组原理图

3 基于遗传算法的混合 Adaboost 算法（Adaboost - GA）

本文所选的弱分类器为上文提到的决策小组。具体算法流程如图 2。

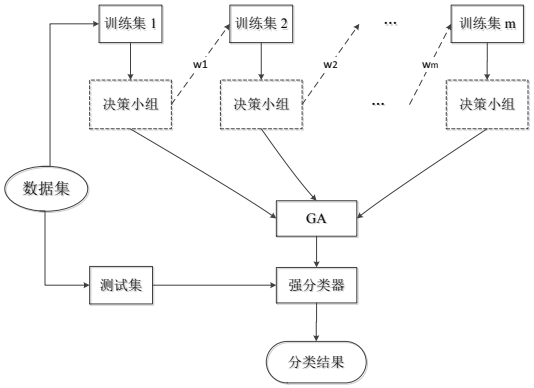

图 2 Adaboost-GA 算法流程图

设 D＝$\{(x_1,y_1),(x_2,y_2),....,(x_N,y_N)\}$ 为训练集。x 是样本的属性，y 为该样本所属的类别。$\omega_{m,i}$ 表示第 m 个弱分类器中第 i 个样本的权重。其中，ω_m 为 n 个样本的权重

集合。$y_m(x)$为弱分类器，$Y(x)$为强分类器，α_m表示第m个弱分类器的话语权，ε_m表示第m个弱分类器的误差，M 为弱分类器的个数。算法步骤：

1）将数据集划分为训练集和测试集。

对于训练集：

2）初始化训练集中 n 个训练样本的权重为 $1/n$。每个样本通过决策小组进行分类，若能正确分类则该样本的权重减小。

3）for $i=1,...,n$：

a）训练弱分类器 $y_m(x)$，最小化权重误差：

$$\varepsilon_m = \sum_{i=1}^{n} \omega_{m,i} I(y_m(x_i) \neq y_i) \quad (1)$$

b）接下来计算该弱分类的话语权：

$$\alpha_m = \ln\{\frac{1-\varepsilon_m}{\varepsilon_m}\} \quad (2)$$

c）更新样本权重。

4）再用遗传算法优化弱分类器的话语权 α_m。

5）得到最后的强分类器：

$$Y(x) = sign(\sum_{j=1}^{m} \alpha_j y_j(x)) \quad (3)$$

对于测试集：

6）将测试集放到训练完毕的强分类器中进行分类。

7）得到分类结果。

4 实验

为了评估 Adaboost-GA 算法的性能，本文对该算法进行了实验分析与仿真，从 UCI 标准分类中数据库中选择五个基因数据集进行实验。实验结果如图 3 所示：

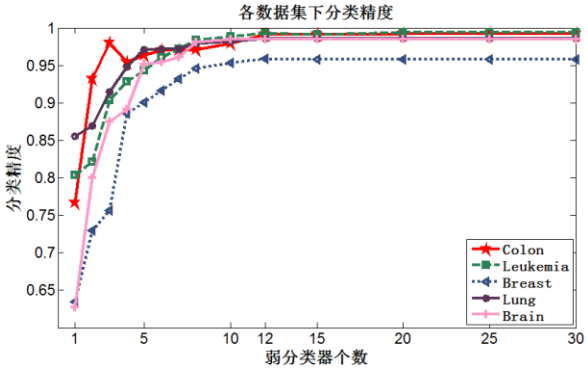

图 3 Adaboost-GA 算法的分类效果

图 3 是各数据集上 Adaboost-GA 算法的分类效果。可知，该算法可以很好地应用在各基因数据集上，并且随着弱分类器个数的增加，分类精度在 10 个弱分类器时基本保持不变，可达到 95%。

图 4 是 Colon 基因数据集上不同算法的分类精度比较图。可以看出，Adaboost-GA 的分类效果远远好于其他算法，且在达到较好分类效果时所需的弱分类器个数最少。同时，Adaboost-GA 的稳定性更好。其中，非集成算法 ELM 也表现出了良好的分类效果，但是稳定性不足。

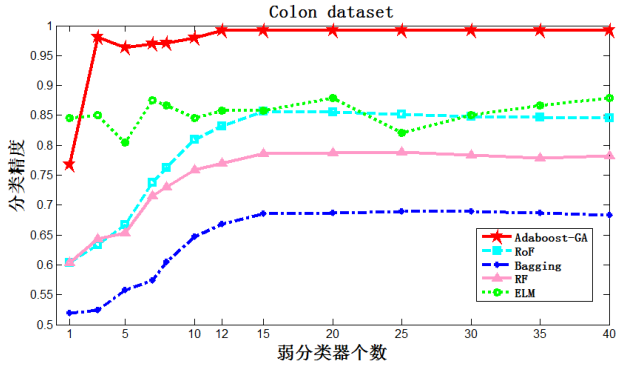

图 4 Colon 数据集的各算法分类精度

将 Adaboost-GA 算法分类精度的平均值与 SVM、DT、KNN 以及 KNN 算法的分类精度进行比较。如表 1，可以看出，文中所提算法的分类精度明显高于其他算法。

表 1 几种算法分类精度比较

数据集 \ 算法	SVM	DT	KNN	NB	Adaboost-GA
Colon	0.919	0.904	0.890	0.727	0.955
Breast	0.826	0.736	0.694	0.626	0.877
Lung	0.785	0.839	0.846	0.635	0.951

5 结论

本文通过改善弱分类器以及优化弱分类器权值的方法来提高 Adaboost 算法的分类精度。经实验证明，本文所提的 Adaboost-GA 算法很好地达到了提高分类精度的效果，且在较小的集成度时就能达到很好的分类效果。但是，本文算法比较复杂，耗时较长。因此，如何减少耗时将是本文下一步的主要研究工作之一。

REFERENCES

[1] 陆慧娟，安春霖，马小平, 等. 基于输出不一致测度的极限学习机集成的基因表达数据分类[J]. 计算机学报, 2013, 36(2):341-348.

[2] Adeniyi D A, Wei Z, Yongquan Y. Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method [J]. Applied Computing & Informatics, 2016, 12(1):90–108.

[3] 王双成，杜瑞杰，刘颖. 连续属性完全贝叶斯分类器的学习与优化[J]. 计算机学报, 2012, 35(10):2129-2138.

[4] Podgorelec V, Zorman M. Decision Tree Learning [M]// Machine Learning Models and Algorithms for Big Data Classification. 2016:1751 - 1754.

[5] Nayak D R, Dash R, Majhi B. Brain MR image classification using two-dimensional discrete wavelet transform and AdaBoost with random forests[J]. Neurocomputing, 2016, 177(C): 188-197.