Tumor MRI classification via Boosting

Mohammed Adib Oumer

Spring ‘22 ECEN 5712 Machine Learning for Engineers

Why Tumor MRI?

- As aspiring researcher, would like to test my knowledge on real life critical applications
- Brain tumor classification plays a niche role in medical prognosis and effective treatment process. Among the fatal diseases, brain cancer is particularly difficult because it is not usually detected until it is too late for prognosis [1].

Why ML?

- Facilitate the detection process.
- Personal motivation: research focused around optimization. Compare to DL results. Testing what classifiers work.

Boosting

- **Algorithm:** AdaBoost (multi-class version), first presented by Freund and Schapire in the year 1997. Most commonly used in binary classification

- Weak Learners: Gaussian Naive Bayes, Multinomial Naive Bayes, Logistic (Softmax) Regression (custom implementations), Decision Tree (scikit)

Boosting

Steps:

- Assign uniform weight to all data points

■

- Find respective values from subset of training data selected based on assigned weight:
  - p(xly) and p(y) for Naive Bayes
  - coefficients in logistic regression (gradient descent)

■

- Get weighted errors

■

- Calculate performance

$$p_i = \frac{1}{m}$$

$$p(y) = \frac{m_y}{m}$$

$$p(\mathbf{x}|y) \sim \mathcal{N}(\hat{\mu}, \hat{\Sigma}^2)$$

$$p(x_i|y) = \frac{N_{yi} + \alpha}{N_y + \alpha m}$$

$$e_j = \sum_{i=1}^{m} \mathbb{1}(\hat{y}_i \neq y_i) \times p_i$$

$$\alpha_j = \ln\left(\frac{1 - e_j}{e_j}\right) + \ln(K - 1)$$

Boosting

Steps (continued):

- Update weights based on performance (two options)
  - Increase for misclassified, keep for correct

$$\mathbf{w}_{j+1} = \mathbf{p}_j \cdot e^{\alpha_j \times \mathbb{1}(\hat{y}_i \neq y_i)}$$

  - Increase for misclassified, decrease misclassified

$$\mathbf{w}_{j+1} = \mathbf{p}_j \cdot e^{-\alpha_j} (correct)$$

$$\mathbf{w}_{j+1} = \mathbf{p}_j \cdot e^{\alpha_j} (misclassified)$$

- Normalize

$$\mathbf{p}_{j+1} = \frac{1}{\mathbb{1}^T \mathbf{w}_{j+1}} \mathbf{w}_{j+1}$$

- Iterate for given number of iterations
- Combine results for final prediction

$$\mathbf{prediction}_{final} = \arg\max_y \sum_{j=1}^{M} \alpha_j \mathbb{1}(\hat{\mathbf{y}}_\mathbf{j} = y)$$

Results:

Result for 64x64 Images		
Iterations = 1	Accuracy(%)	
	Training	Test
Single Gaussian Naive Bayes	62	58
AdaBoost Gaussian Naive Bayes	62	59
AdaBoost Multinomial Naive Bayes	54	49
Single Logistic Regresssion	66	61
AdaBoost Logistic Regression	70	67
Adaboost Decision Tree (scikit)	60	55

Results:

Result for 64x64 Images		
Iterations = 20	Accuracy(%)	
	Training	Test
AdaBoost Gaussian Naive Bayes (elapsed iteration 11)	71	69
AdaBoost Multinomial Naive Bayes (elapsed 19)	55	56
AdaBoost Logistic Regression	77	72
Adaboost Decision Tree (scikit)	72	66

Result for 64x64 Images		
Iterations = 50	Accuracy(%)	
	Training	Test
AdaBoost Gaussian Naive Bayes (elapsed iteration 11)	71	69
AdaBoost Multinomial Naive Bayes (elapsed 19)	55	56
AdaBoost Logistic Regression	-	-
Adaboost Decision Tree (scikit)	79	73

Observations & Conclusions:

- Error = Bias (underfitting) + Variance (overfitting) + ~~Irreducible (inherent)~~
- Naive Bayesian classifier and Logistics Regression not ideal for boosting
- Both are stable classifiers with low variance and high bias [2]
- Decision tree is a classifier with high variance and low bias
- AdaBoost immune to overfitting with weaker learners

Future works:

- Exploring other classifiers with high variance and low bias classifiers: K-nearest neighbours and Support Vector Machine (SVM) with OVR
- Other Boosting Algorithms: Gradient Boosting, LPBoost, QPBoost

References

[1] A. Veeramuthu, S. Meenakshi, G. Mathivanan, K. Kotecha, J. Saini, V. Vijayakumar, and V. Subramaniyaswamy, “MRI brain tumor image classification using a combined feature and image-based classifier,” *Frontiers in Psychology*, vol. 13, 2022.

[2] Ting, K.M. and Zheng, Z. (2003), A Study of AdaBoost with Naive Bayesian Classifiers: Weakness and Improvement. Computational Intelligence, 19: 186-200.

And class notes

Thank You!