Trabalho prático 2
Aprendizado de Máquina

Gustavo Penha
DCC, UFMG
guzpenha@dcc.ufmg.br

1. IMPLEMENTAÇÃO DO ADABOOST

Nesta seção serão descritas as principais decisões de implementação, assim como a maneira que cada um dos componentes do algoritmo de classificação funciona.

1.1 Weak Learner

O algoritmo de aprendizado base utilizado é uma *Decision Stump*, que gera modelos "fracos" em cima de problemas de classificação binaria e dados categóricos. As funções que este método aprende são cortes nas dimensões do dado de entrada, sendo que cada dimensão do dado gera dois cortes possíveis: um na qual a predição é a própria variável categórica e o outro na qual a predição é o inverso dela. Além disso, o modelo pode fazer predições constantes 0 ou 1.

A implementação realizada possui uma classe chamada *CategoricalStump()*, que possui dois métodos: *fit()* e *predict()*. O primeiro calcula o melhor corte possível nos dados levando em conta o peso de cada entrada (*sample_weight*) no calculo do erro, ajustando o melhor modelo aos dados. Já o segundo método faz as predições baseando-se no melhor corte aprendido.

1.2 Adaptive Boosting Classifier

O algoritmo de ***AdaBoost*** implementado utiliza N classificadores fracos (*Weak Learners*) dando pesos para cada um desses estimadores base de acordo com o quanto cada um deles erra nas entradas. O peso (α) de cada classificador base (H_x) e os pesos de cada instancia seguem as formulas aprendidas em sala de aula, assim como a predição é dada por: $sign(\alpha_0 * H_0(X) + \alpha_1 * H_1(X) + ... + \alpha_N * H_N(X))$.

A implementação criada segue o mesmo padrão do *DecisionStump*, já que a classe *AdaBoostCategoricalClassifier()* implementa dois metodos: *fit()* e *predict()*. O primeiro deles realiza o ajuste de *n_estimators* classificadores base (*DecisionStumps*), calculando e salvando iterativamente o peso α_x e o classificador H_x de acordo com as formulas do AdaBoost [1].

2. AVALIAÇÃO EXPERIMENTAL

Nesta seção os resultados da avaliação experimental do trabalho prático são descritos. Para avaliar o algoritmo criado, foi utilizado o dataset TicTacToe[1], utilizando a acurácia como métrica de avaliação.

[1]https://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame

2.1 Sensitividade em relação ao numero de estimadores

Para avaliar como o algoritmo de aprendizado se comporta ao aumentarmos o numero de estimadores base, utilizamos o procedimento de validação cruzada com k=5. Aumentar a quantidade de *weak learners* base no algoritmo leva a uma melhor acurácia média, como vemos na Figura 1.

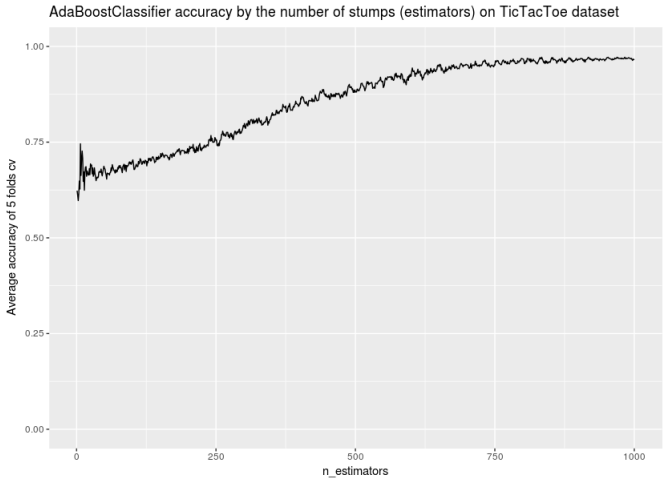

Figure 1: Sensitividade do algoritmo ao aumentar o número de estimadores fracos (decision stumps) no *ensemble*.

3. REFERENCES

[1] Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.