

Método Gradient Boosting

Samuel Hericles

Universidade Federal do Ceará - Campus Sobral
Programa de Educação Tutorial(PET)

27 de Agosto de 2019

- Palavras-chave : Machine learning,junções de métodos e otimização.



Sumário

- 1 Introdução
 - História
 - Definição
- 2 Tipos de métodos ensemble
 - Stacking
 - Bagging
 - Bosting
- 3 Gradiente Boosting
 - Definição
 - Algoritmo
 - Pontos relevantes
 - Implementação
 - Aplicações
- 4 Referências



História

- Dasarathy e Sheela(1979) : "*A Composite Classifier System Design : Concepts and Methodology*[1];
 - Sistema de classificadores composto
 - Classificadores lineares e NN(Nearest neighbor);
 - Dados do IRIS.
- Breiman(1996) : "*Bagging predictors*"[2];
 - "*bootstrap aggregating*";
 - Aumentar a precisão do método.



Métodos Ensemble

- Métodos em conjunto ;
- Obter melhor desempenho ;
- Flexibilidade para ajustar a mais de um modelo.



Métodos Ensemble

- Tipos de métodos ensemble[3][4] :
 - Stacking ;
 - Bagging ;
 - Boosting.



Stacking

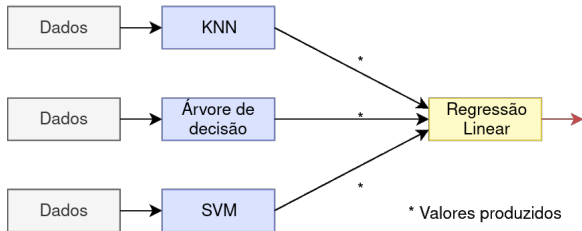


FIGURE – Imagem ilustrativa do algoritmo stacking.
Fonte - Produzida pelo o autor.

Bagging

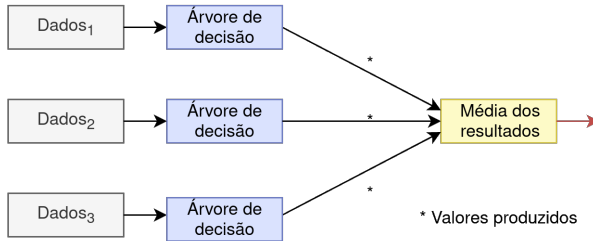


FIGURE – Imagem ilustrativa do algoritmo bagging.
Fonte - Produzida pelo o autor.



Bosting

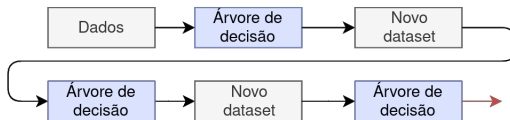


FIGURE – Imagem ilustrativa do algoritmo bosting.
Fonte - Produzida pelo o autor.



Gradiente Boosting

- Algoritmo de árvore de regressão ;
- Para problemas de regressão e classificação[5] ;
- Aumento na precisão da previsão do modelo[6] ;
- Cada modelo tem o objetivo de corrigir os erros dos modelos anteriores[7].



Algoritmo

Seja J o número de folhas de cada árvore em que qualquer uma delas divide um espaço de entrada em regiões $R_{1m}, R_{2m}, \dots, R_{jm}$ e prediz uma constante b_{jm} para região R_{jm} . Logo a função da árvore de regressão é :[7]

$$g_m(x) = \sum_{j=1}^J b_{jm} h(x), \quad (1)$$

sabendo que,

$$h(x) = \begin{cases} 1, & \text{se } x \in R_{jm} \\ 0, & \text{caso contrário} \end{cases} \quad (2)$$



Algoritmo

Usando a função de árvore de regressão $g_m(x_i)$ na função de *Gradient Boosting* genérica $f_m(x)$, temos o modelo de atualização do gradiente descendente :

$$f_m(x) = f_{m-1}(x) + \rho_m g_m(x), \quad (3)$$

considerando ρ_m como,

$$\rho_m = \underset{\rho}{\operatorname{argmin}} \sum_{i=1}^n L(y_i, f_{m-1}(x) + \rho g_m(x)). \quad (4)$$



Algoritmo

Logo, substituindo (4) em (3), temos :

$$f_m(x) = f_{m-1}(x) + \sum_{j=1}^J p_m b_{jm} h(x) \quad (5)$$

usando ρ_{jm} em cada região de R_{jm} , b_{jm} pode ser desconsiderado. Portanto, a regra de atualização do modelo é portanto :

$$f_m(x) = f_{m-1}(x) + \sum_{j=1}^J p_m h(x), \quad (6)$$

$$\rho_m = \underset{i=1}{\operatorname{argmin}}_p \sum_{i=1}^n L(y_i, f_{m-1}(x) + \sum_{j=1}^J p_m h(x)). \quad (7)$$



Algoritmo

O ajuste excessivo pode ser controlado através de parâmetro de contribuição de cada árvore, chamado de fator **R** (taxa de aprendizado) :

$$f_m(x) = f_{m-1}(x) + R \sum_{j=1}^J p_m h(x) \quad (8)$$



Pontos relevantes

- Parametros do Gradient Boosting :
 - Número de nós em cada árvore ;
 - Profundidade da árvore(número de iterações) ;
 - Taxa de aprendizagem ou fator **R** .
- Análise sobre os parâmetros de **R** e **M** ;
- Aumentar o número de modelos ;
- Complexidade da árvore.



Implementação - Pseudo-Código

Inicie $f_0(x)$ para ser uma constante, $f_0(x) = \operatorname{argmin}_p \sum_{i=1}^N L(y_i, p)$

Para $m = 1$ até M **faça**

Para $i = 1$ até n **faça**

Calcule o gradiente negativo

$$z_{im} = - \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f=f_{m-1}}$$

fim

Ajuste uma árvore de regressão $g_m(x)$ para prever os destinos z_{im} das covariáveis x_i para todos os dados de treinamento.

Calcule um tamanho da etapa de descida de gradiente como:

$$\rho_m = \operatorname{argmin}_\rho \sum_{i=1}^n L(y_i, f_{m-1}(x_i) + \rho g_m(x_i))$$

Atualize o modelo com

$$f_m(x) = f_{m-1}(x) + \rho_m g_m(x)$$

fim

Saída: $f_M(x)$



Implementação - Python

```
def prod(x):  
    y = x / 2 + (x // 10) % 2 * 20 * x / 5 + np.random.random() * 10  
    return y  
  
#Definindo entrada e saída para o teste com o modelo  
x = np.arange(0, 60)  
y = prod(x)  
x = pd.DataFrame({'x': x})  
  
# Cria o modelo de gradient boosting, definidos os parâmetros  
params = {  
    'n_estimators': 10,  
    'max_depth': 5,  
    'learning_rate': 1,  
    'criterion': 'mse'  
}  
  
gradient_boosting_regressor = ensemble.GradientBoostingRegressor(**params)  
gradient_boosting_regressor.fit(x, y)
```



Aplicações

- Mapear sinais EMG[8];
- Classificador de atividades físico[8];
- Classificação de texto com modelos esparsos[8];
- Previsão da presença de espécies em florestas[9].



Referências I

- [1] BELUR V. DASARATHY and BELUR V. SHEELA.
A Composite Classifier System Design : Concepts and Methodology, volume 67.
IEEE XPLOER, [https ://ieeexplore.ieee.org/document/1455590/](https://ieeexplore.ieee.org/document/1455590/), 1979.
- [2] Leo Breiman.
Bagging Predictors, volume 24.
Kluwer Academic Publisher, Boston, 1993.
- [3] Ensemble learning.
https://en.wikipedia.org/wiki/Ensemble_learning.
Acesso : 20/08/2019.
- [4] Machine Learning for Everyone.
https://vas3k.com/blog/machine_learning/.
Acesso : 20/06/2019.
- [5] Gradient boosting.
https://en.wikipedia.org/wiki/Gradient_boosting.
Acesso : 20/08/2019.



Referências II

- [6] Thomas G. Dietterich.
An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees : Bagging, Boosting, and Randomization, volume 40.
Kluwer Academic Publisher, Manufactured in The Netherlands, 2000.
- [7] Jian Ma Hongbing Jiang Gang Wang, Jinxing Hao.
A comparative assessment of ensemble learning for credit scoring, volume 38.
ScienceDirect, www.elsevier.com/locate/eswa, 2011.
- [8] Alexey Natekin and Alois Knoll.
Gradient boosting machines, a tutorial.
frontiers in NEUROBOTICS, 2013.
- [9] Jock A. Blackard Tracey S. Frescino Niklaus E. Zimmermann Thomas C. Edwards Jr. Gretchen G. Moisen, Elizabeth A. Freeman.
Predicting tree species presence and basal area in Utah : A comparison of stochastic gradient boosting, generalized additive models, and tree-based methods, volume 199.
ScienceDirect, www.elsevier.com/locate/ecolmodel, 2006.



Referências III

- [10] Machine Learning Introduction : A Comprehensive Guide.
<https://towardsdatascience.com/machine-learning-introduction-a-comprehensive-guide-af6712cf68a3>.
Acesso : 20/06/2019.
- [11] Jerome H. Friedman.
Greedy function approximation : A gradient boosting machine, 1999.
- [12] Ali Haghani Yanru Zhang.
A gradient boosting method to improve travel time prediction.
ScienceDirect, www.elsevier.com/locate/trc, 2016.
- [13] Loss function.
https://en.wikipedia.org/wiki/Loss_function.
Acesso : 20/08/2019.

