Árvore de Modelos Lineares Generalizados: um algoritmo flexível para dados contínuos positivos

Alberto Rodrigues[1] , Tibérius O. Bonates[2] e Juvêncio Santos Nobre[2]

[1] Instituto de Matemática e Estatística - Universidade de São Paulo
albertor@ime.usp.br
[2] Departamento de Estatística e Matemática Aplicada - Universidade Federal do Ceará
{tb,juvencio}@ufc.br

Resumo Um algoritmo de aprendizagem supervisionada, Árvore de Modelos Lineares Generalizados (AMG) foi desenvolvido e implementado envolvendo dois métodos clássicos de estatística e aprendizagem supervisionada, os modelos lineares generalizados e árvore de regressão. Para previsão de variáveis contínuas, já existem abordagens que trabalham com árvores de modelos, ou seja uma árvore de decisão que contém expressões lineares, estimadas por regressão linear múltipla. Neste artigo desenvolvemos um algoritmo que abrange mais opções de predição, construção da árvore, obtenção de variáveis relevantes, hiperparâmetros e simplificação da árvore construída. O algoritmo é composto de algumas etapas principais: (1) são separados subconjuntos dos dados, (2) são ajustados diferentes modelos para cada subconjunto, (3) é realizada uma seleção de variáveis com o método *backward*, (4) é feita uma simplificação da árvore. Comparações foram realizadas em dados reais com diversos algoritmos consolidados na literatura e mostramos que a AMG possui boa eficácia e robustez. Esse algoritmo pode ser utilizado por especialistas com o intuito de visualização, interpretação das variáveis, predição e análise de dados do mundo real.

Keywords: Modelos Lineares Generalizados · Modelos Baseados em Partição Recursiva · Modelos Baseados em Árvore · Árvore de Modelos

1 Introdução

Um dos algoritmos mais conhecidos de aprendizagem de máquina é a árvore de decisão/regressão, algumas implementações são: CART [3] e C4.5 [11], especificamente no caso de regressão às árvore predizem médias ou medianas, ou seja, para um dado subconjunto a predição é sempre a mesma, o que torna uma predição ruim nesse aspecto.

Outro algoritmo baseado em árvores é a árvore de modelos para regressão que é um algoritmo que combina árvore de regressão e regressão linear múltipla nos nós, em que a predição é fornecida por uma expressão linear, o que torna a

predição mais eficaz, porém só há um único modelo teórico associado. Algumas implementações são: M5 [12], M5'[13]. Outros algoritmos baseados em árvores de modelos foram estudados com outros tipos de modelos de regressão, por exemplos, uma abordagem bayesiana dos modelos lineares generalizados [6], modelos não-paramétricos e modelo Poisson[5], modelo linear múltiplo[2] e modelos polinomiais[4] .

Existem árvores de modelos para classificação em que a ideia é similar, um algoritmo que aborda essa ideia é a Árvore de Modelos Logística[9], que combina árvore de decisão e regressão logística para prever a probabilidade de sucesso de uma das classes. Existem outros trabalhos nesse sentido como em [8].

Uma abordagem que incorpora a ideia geral de partição por meio de árvores, que se baseia em utilizar qualquer modelo estatístico paramétrico é o método de modelos baseados em partição recursiva [14]. Este tipo de método utiliza de um modelo $\mathcal{M}(Y, \theta)$, que são compostos por Y, que é o vetor de variáveis aleatórias e um vetor de parâmetros $\theta \in \Theta$. Considerando n observações de Y, os parâmetros são obtidos por meio de uma classe geral de estimadores chamada de estimadores do tipo M, que se baseia em minimizar uma função genérica $\Psi(Y, \theta)$, assim obtemos as estimativas de θ

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{i=1}^{n} \Psi(Y_i, \theta) \tag{1}$$

Os estimadores de máxima verossimilhança são um caso particular, considerando $\Psi(Y, \theta)$ como a log-verossimilhança negativa.

Uma alternativa flexível aos modelos baseados em partição recursiva é a AMG, em que adaptamos a forma da construção da árvore, ajustamos o melhor modelo linear generalizado em cada subconjunto para termos melhores opções de predição, obtemos variáveis relevantes através do método *backward*, adaptamos hiperparâmetros capazes de evitar *overfitting* e de se obter a árvore mais rapidamente e ajustamos maneiras de simplificação da árvore, para evitar *overfitting* dos dados.

A principal diferença que nos distingue dos trabalhos e artigos anteriores é devido ao maior poder preditivo por conta de uma maior quantidade de modelos a serem estimados e do método que tenta evitar custo computacional elevado ao estimar esses modelos.

Uma AMG tem como objetivo melhorar a precisão da variável resposta do que uma árvore de modelos tradicional que utiliza somente regressão linear múltipla e compartilhar uma alternativa promissora em relação a outros algoritmos que apresentam essa mesma ideia.

A AMG é um algoritmo de aprendizagem de máquina supervisionado que se baseia em particionar um dado conjunto de dados com o intuito minimizar a variabilidade das partições utilizando o desvio padrão local, ou seja, o algoritmo divide o conjunto de dados em subconjuntos de dados similares entre si a fim de melhorar a previsão da variável resposta. Dado o formato de uma árvore, queremos modelar $E[Y_i/X_i]$, $\forall i = 1, \cdots, n$, por meio de um modelo paramétrico

mais sofisticado, quando temos evidências de que um modelo ajustado a todo o conjunto de dados não é adequado.

No restante deste artigo, as seções são seguidas por uma ideia geral do algoritmo proposto, particionamento do conjunto de dados, modelos particionados, seleção de variáveis, hiperparâmetros, simplificação da árvore, casos particulares, predição, resultados e conclusão.

2 Algoritmo

A ideia básica de uma AMG é de que subconjuntos de dados similares tenham um modelo de regressão que prediz melhor comparado a algum modelo de regressão ajustado com todos os dados, pois para o conjunto de dados inteiro pode ser que um único modelo paramétrico não sejam adequado, porém em algum subconjunto dos dados possa ser bem ajustado. Então, o objetivo na construção de uma AMG é fazer partições a fim de realizar uma previsão adequada para uma nova observação, dada que seja de um determinado subconjunto dos dados de treinamento e a partir daí realizar a previsão com base na equação de regressão desse grupo.

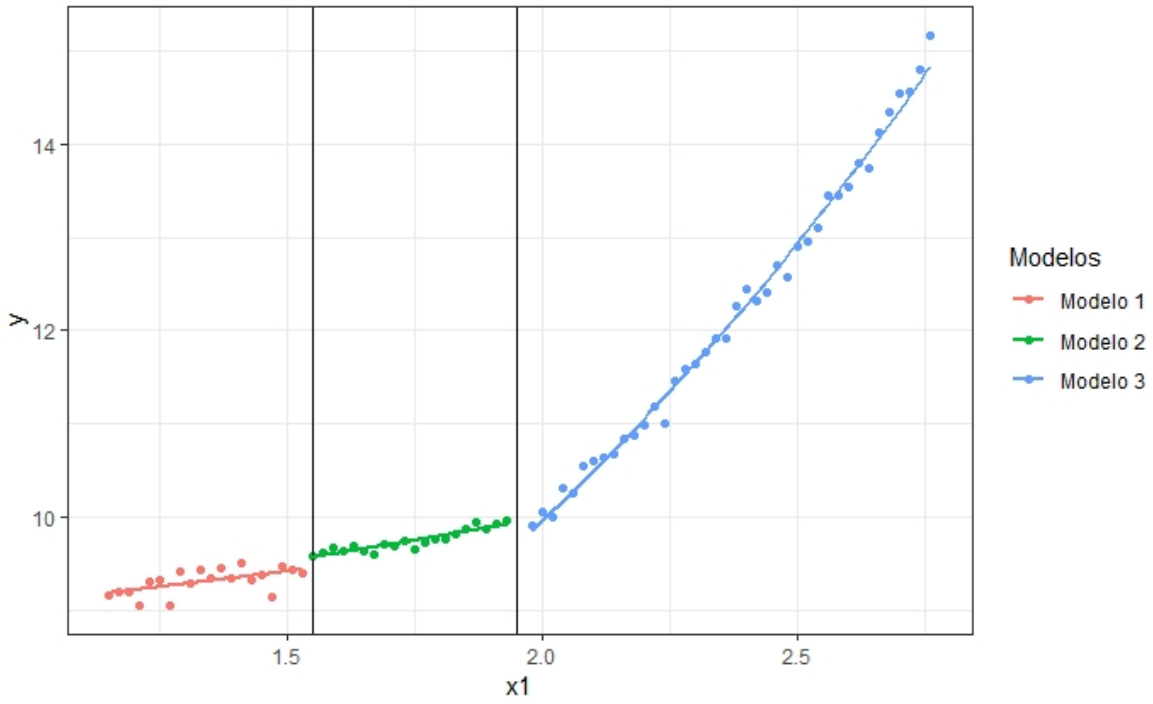
Os passos do treinamento e previsão de uma AMG são descritos a seguir:

1. São separados subconjuntos dos dados para minimizar a variabilidade da variável resposta.
2. São ajustados diferentes modelos de regressão para cada um dos subconjuntos do passo anterior. O modelo que mais se ajusta a cada subconjunto é escolhido para ser o modelo final.
3. Para cada modelo final nos subconjuntos é realizado uma seleção de variáveis usando o método backward, selecionando variáveis significativas para o modelo.
4. É realizada uma simplificação da árvore, com o objetivo de evitar *overfitting*.
5. Para realizar a previsão é feita uma verificação de quais nós terminais as observações pertencem.
6. Com base na equação de regressão do modelo final de cada nó terminal, as observações são preditas.

Exemplo: Foram feitas duas partições do conjunto de dados por completo, que por sua vez só possuía uma variável explicativa (x_1) e uma variável resposta (y). A primeira partição realizada foi $x_1 \leq 1.95$ contra $x_1 > 1.95$ e a segunda foi $x_1 \leq 1.95$ e $x_1 \leq 1.55$ contra $x_1 \leq 1.95$ e $x_1 > 1.55$, resultando em três subconjuntos. Para cada subconjunto foram ajustados vários modelos de regressão e o modelo mais adequado foi selecionado para a árvore.

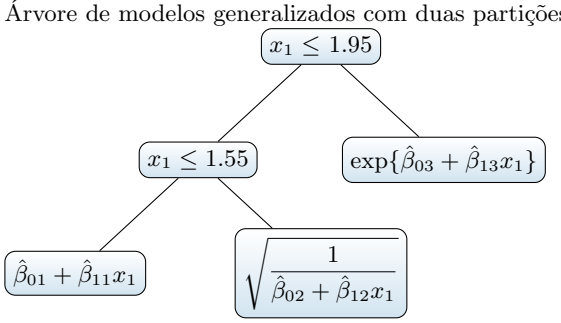
O gráfico com os subconjuntos é dado a seguir:

Figura 1: Três modelos lineares generalizados ajustados em subconjunto dos dados.



A árvore construída a partir desses dados, resultaria em uma árvore com três subconjuntos dada a seguir:

Árvore de modelos generalizados com duas partições.



Em que $\hat{\beta}_{0j}$ e $\hat{\beta}_{1j}$, j=1,2,3, são estimativas de máxima verossimilhança dos parâmetros dos modelos de regressão para cada subconjunto.

3 Particionamento do conjunto de dados

O primeiro passo para construção do algoritmo é separar subconjuntos dos dados, de maneira que obtenhamos modelos mais robustos para cada subconjunto.

Essas separações são obtidas a partir dos valores das variáveis explicativas que fornecem subconjuntos similares. A AMG é um algoritmo recursivo, ou seja, o procedimento de achar um valor separador de alguma variável depende dos subconjuntos gerados pelo nó pai. Assim, para cada subconjunto da árvore é realizada a verificação da métrica dada na equação 2.

A métrica utilizada para separação é a mesma utilizada em [12]. O motivo de sua utilização é que subconjuntos que tiverem desvio padrão menor da variável resposta em relação ao conjunto de dados inteiro terá uma melhor performance do modelo final. Dessa forma, o subconjunto será mais facilmente ajustado por conta de estar menos disperso e consequentemente os parâmetros do modelo final de cada subconjunto serão melhores estimados.

As partições do espaço das variáveis explicativas S_1 são os conjuntos $S_1, S_2, \cdots, S_k$, de tal forma que os conjuntos são gerados por

$$\begin{cases} S_i = S_{2i} \cup S_{2i+1}, \forall i = 1, \cdots, \lfloor k/2 \rfloor \\ S_{2i} \cap S_{2i+1} = \emptyset \end{cases}$$

Cada nó interno da árvore é realizada uma busca exaustiva para encontrar as melhores partições de tal forma que maximize a métrica

$$V(S_i) = D.P(S_i) - \frac{|S_{i1}|}{|S_i|} D.P(S_{i1}) - \frac{|S_{i2}|}{|S_i|} D.P(S_{i2}), \forall i = 1, \cdots, \lfloor k/2 \rfloor \qquad (2)$$

em que $|S_{i1}|$, $|S_{i2}|$ e $|S_i|$ são os números de observações que pertencem ao subconjuntos S_{i1}, S_{i2} e S_i, respectivamente, $D.P(S_i)$, $D.P(S_{i1})$ e $D.P(S_{i2})$ são os desvios padrões da variável resposta dos subconjuntos S_i, S_{i1} e S_{i2} respectivamente.

O procedimento para encontrar a melhor separação por meio da métrica $V(S_i)$ de um subconjunto é fazendo uma busca exaustiva de subconjuntos S_{i1} e S_{i2} através das variáveis explicativas do conjunto de dados.

O processo de determinação dos subconjuntos S_{i1} e S_{i2} é mostrado a seguir.

Variáveis explicativas numéricas

1. Para cada variável explicativa numérica x são observados seus valores únicos e são ordenados, denotaremos esse conjunto de valores de A_x
2. Para cada $a_j \in A_x, j \in 1, \cdots, \#A_x$-1, é calculado $R_j = \dfrac{a_j + a_{j+1}}{2}$
3. Os conjuntos S_{i1} e S_{i2} são formados de tal forma
   $S_{i1} = \{S_i : x \leq R_j\}$
   $S_{i2} = \{S_i : x > R_j\}$

Variáveis explicativas categóricas

1. Para cada variável explicativa categórica x, seus fatores são observados e é denotado por A_x
2. Para cada $a_j \in A_x, j \in 1, \cdots, \#A_x$ são formados os conjuntos S_{i1} e S_{i2} de tal forma
   $S_{i1} = \{S_i : x = R_j\}$
   $S_{i2} = \{S_i : x \neq R_j\}$

4 Modelos particionados

Após construída a árvore, modelos lineares generalizados(MLGs) são ajustados a todos os subconjuntos. A distribuição de probabilidade de Y(variável resposta) tem que pertencer à família exponencial.

$$f(y;\phi,\theta) = exp\{a(\phi)^{-1}[y\theta - b(\theta)] + c(y,\phi)\}\ \forall \phi,\theta \in \Theta \tag{3}$$

De forma que a relação entre a esperança de Y esteja relacionado com o preditor linear através de $g(\mu) = x^T\beta$.

As distribuições utilizadas pela AMG dessa classe de distribuições são: normal, gama e normal inversa com funções de ligação identidade, inversa e logarítmica. Esses modelos são ajustados a fim de realizar previsões adequadas para subconjuntos específicos.

4.1 Estimação dos parâmetros do modelo

A estimação dos parâmetros do modelo é realizada por meio do método de máxima verossimilhança. Aqui será considerada uma amostra da variável resposta dada por $Y_1, \cdots, Y_n$, independentes. Os passos da estimação são:

1. Para cada subconjunto $S \in \{S_1, S_2, \cdots, S_k\}$, são ajustados modelos de regressão $\mathcal{M}_j(Y,\beta) \in \mathcal{G}$
2. Os parâmetros β de cada modelo $\mathcal{M}_j(Y,\beta)$ são ajustados através de:

$$\hat{\beta} = \underset{\beta\in\Theta}{\operatorname{argmin}} \sum_{i=1}^{n} \Psi_j(y_i,\beta) \underset{S}{I(i)} \tag{4}$$

Para mais detalhes sobre a estimação dos parâmetros, consulte [10],[7].

4.2 Seleção de modelos recursiva

Devido ao possível alto custo computacional, pelo motivo de se ajustar vários modelos em cada nó é realizado uma seleção de modelos, em que se não avalia todos os modelos em todos os nós. O método de seleção de modelos recursiva, que tenta resolver esse problema é dado a seguir:

1. No nó raiz, todos os modelos são ajustados e para cada modelo é atribuído um erro que chamaremos de ε_i, i=1,2,...,m.
2. Para cada modelo no nó raiz, um peso é atribuído a cada modelo, que é dado por $\dfrac{1}{\varepsilon_i}$. Quanto menor o erro maior será o peso do referido modelo para os próximos nós. O erro pode ser qualquer métrica conhecida que possa ser minimizada, por padrão é usado o erro médio absoluto.
3. Um hiperparâmetro entre 0 e 1 (que pode ser atualizado) é usado para selecionar uma proporção dos modelos que obtiveram maiores pesos para o próximo nó.
4. Esse processo é repetido em todos os nós até que alcance pelo menos um hiperparâmetro de parada.

5 Seleção de variáveis

A interpretabilidade é uma das grandes vantagens de uma AMG, assim como algoritmos baseados em modelos paramétricos. O objetivo dessa etapa do algoritmo é fazer um pré-processamento inicial, que é descartar variáveis explicativas irrelevantes, que não contribuam significativamente para o modelo final de um dado subconjunto. Estamos interessados em atualizar o modelo final para que tenhamos previsões adequadas com apenas um subconjunto de variáveis.

A seleção de variáveis da AMG é baseada na minimização do critério AIC [1], que é uma função baseada na log-verossimilhança negativa penalizada pela quantidade de parâmetros do modelo paramétrico. A obtenção do melhor conjunto de variáveis explicativas com base no AIC é realizada por meio da regressão *backward stepwise* de cada subconjunto.

Dessa forma, responsáveis pela tomada de decisão podem analisar as variáveis significativas do modelo e verificar qual o caminho da árvore que levaram a predição obtida.

6 Hiperparâmetros da árvore

Os hiperparâmetros do algoritmo são regularizadores que alteram o funcionamento do algoritmo. São variáveis que afetam todo o processo de treinamento da árvore, como o critério de parada. Os hiperparâmetros mais importantes são descritos a seguir:

6.1 Profundidade máxima

É o número máximo de níveis que a árvore pode ter. É um hiperparâmetro essencial para se evitar subconjuntos específicos que ajudam a ocorrência de *overfitting*.

6.2 Número mínimo de observações em nós

O número mínimo de observações em nós são os principais regularizadores para se evitar *overfitting*, pois através deles a partir de um certo ponto a árvore para de crescer para subconjuntos que ajudem a ocasionar *overfitting*.

6.3 Redução percentual mínima do desvio padrão

É um hiperparâmetro que verifica se um subconjunto está menos disperso para se obter melhores estimativas dos modelos. Isso é realizado da seguinte forma: $D.P(S_i) \leq c * D.P(S_1), \forall i = 2, \cdots, \lfloor k/2 \rfloor, \forall c \in (0, 1)$

6.4 Porcentagem de modelos

Proporção dos modelos a serem selecionados em cada nó da árvore, que será utilizado na seleção de modelos recursiva (4.2). É um hiperparâmetro que pode selecionar muitos modelos e ter mais opções de predição se estiver próximo de 1, pode optar por se ter um treinamento mais rápido selecionando menos modelos se estiver próximo de 0 ou tentar obter um meio termo entre essas duas alternativas.

7 Simplificação da árvore

O processo de simplificação da árvore é essencial para evitar *overfitting*, pois uma árvore pode ter subconjuntos muito específicos por ter uma profundidade muito alta ou número baixo de observações em um subconjunto. Este procedimento é realizado depois que cada subconjunto possui um modelo final associado. A simplificação é realizada de baixo para cima, da mesma forma de uma árvore de decisão. O objetivo é tornar a árvore mais simples e mais eficiente para obtermos melhores predições e interpretações viáveis.

7.1 Método com remoção de nós

Este método requer um conjunto de poda, que será usado exclusivamente para a simplificação da árvore. O conjunto de poda é selecionado através do conjunto de treinamento, que é obtido separando um percentual do conjunto de treinamento.

Essa abordagem leva em consideração a possibilidade de que algum subconjunto não possua nenhuma observação do conjunto de poda. A quantidade de observações do conjunto de poda do subconjunto S_i será denotado por $\#S_i, \forall i = 1, \cdots, \lfloor k/2 \rfloor$ e E(.) é o erro a ser computado, pode ser qualquer métrica conhecida que possa ser minimizada, por padrão é usado o erro médio absoluto. Este método de simplificação é descrito a seguir:

1. Para todo $S_i, \forall i = \lfloor k/2 \rfloor, \lfloor k/2 \rfloor - 1, \cdots, 1$, é verificado se $\#S_{2i} \geq 1$ e $\#S_{2i+1} \geq 1$
2. Calculamos o E(S_i),E(S_{2i}) e E(S_{2i+1}) baseado no conjunto de poda.
3. A simplificação da árvore é realizada através das seguintes condições:

$$\begin{cases} S_i = S_i, S_{2i} = \emptyset, S_{2i+1} = \emptyset & \text{se } E(S_i) \leq \dfrac{|S_{2i}|}{|S_i|}E(S_{2i}) + \dfrac{|S_{2i+1}|}{|S_i|}E(S_{2i+1}) \\ S_i = S_i, S_{2i} = S_{2i}, S_{2i+1} = S_{2i+1} & \text{se } E(S_i) > \dfrac{|S_{2i}|}{|S_i|}E(S_{2i}) + \dfrac{|S_{2i+1}|}{|S_i|}E(S_{2i+1}) \end{cases}$$

7.2 Método com correção

Este método baseia-se somente no conjunto de treinamento, os erros são calculados e multiplicados por uma correção, a razão para isso é que geralmente o erro é subestimado no conjunto de treinamento. Essa correção é a mesma que a árvore de modelos [12]. Denotaremos E(S_i^*)=E(S_i)$\left(\dfrac{n_i + p_i}{n_i - p_i}\right)$, em que n_i é o número de observações e p_i é o número de parâmetros do modelo final do subconjunto $S_i, \forall i = 1, \cdots, k$.

1. Para todo $S_i, \forall i = \lfloor k/2 \rfloor, \lfloor k/2 \rfloor - 1, \cdots, 1$, são calculados $\mathrm{E}(S_i^*)$, $\mathrm{E}(S_{2i}^*)$ e $\mathrm{E}(S_{2i+1}^*)$ do conjunto de treinamento.
2. A simplificação da árvore é realizada através das seguintes condições:

$$\begin{cases} S_i = S_i, S_{2i} = \emptyset, S_{2i+1} = \emptyset & \text{se } E(S_i^*) \leq \dfrac{|S_{2i}|}{|S_i|} E(S_{2i}^*) + \dfrac{|S_{2i+1}|}{|S_i|} E(S_{2i+1}^*) \\ S_i = S_i, S_{2i} = S_{2i}, S_{2i+1} = S_{2i+1} & \text{se } E(S_i^*) > \dfrac{|S_{2i}|}{|S_i|} E(S_{2i}^*) + \dfrac{|S_{2i+1}|}{|S_i|} E(S_{2i+1}^*) \end{cases}$$

8 Caso particulares

Em alguns casos o algoritmo desenvolvido pode ter comportamento similar ou igual a modelos consolidados, como os modelos lineares generalizados, árvore de regressão e árvore de modelos.

8.1 Modelos lineares generalizados

Existem duas ocasiões em que um modelo linear generalizado é um caso particular de uma AMG. O primeiro caso é quando não houver nenhum separador em que $V(S_i)$, dado em (2) que seja menor ou igual que o desvio padrão da variável resposta, neste caso o melhor modelo linear generalizado ajustado ao conjunto de dados completo será o único modelo a ser usado na AMG. O segundo caso é quando a AMG tem pelo menos três modelos ajustados, ou seja, pelo menos uma partição do conjunto de dados, mas com algum dos dois tipos de poda, dados em (7) pode se reduzir também a um único modelo linear generalizado.

8.2 Árvore de regressão

Para um caso em que desejamos uma predição constante para todos os subconjuntos podemos considerar um estimador fortemente consistente para esperança de Y, que pela lei forte dos grandes números é $\bar{Y}$.

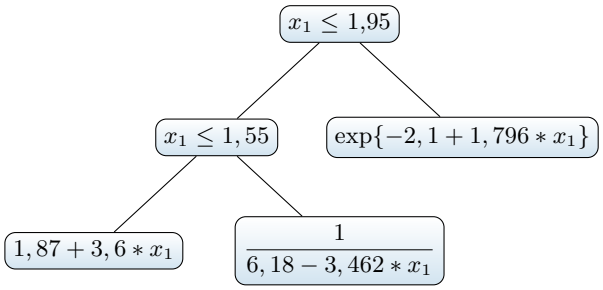
8.3 Árvore de modelos

Intuitivamente a árvore de modelos com base em regressão linear múltipla pode ser considerada um caso particular pelo simples motivo de considerar um único modelo linear generalizado com distribuição normal e função de ligação identidade para todos os nós, com o estimador dado por $\hat{\beta} = (X^T X)^{-1} X^T Y$ para cada subconjunto.

9 Predição

Para realizar a previsão é feita uma verificação de quais nós terminais as observações a serem preditas pertencem. Com base na equação de regressão do modelo final de cada nó terminal, as observações são preditas.

Exemplo: Considerando a seguinte árvore, as observações seriam preditas da seguinte forma:



Para um nova observação com variável explicativa de x_1, temos as predições:

x_1	Predição
2,4	$\exp\{-2,1 + 1,796 * 2,4\} = 9,12$
1,75	$(6{,}18\text{-}3{,}462{*}1{,}75)^{-1} = 8,23$
3,31	$\exp\{-2,1 + 1,796 * 3,31\} = 46,75$
1,55	$1{,}87{+}3{,}6{*}1{,}55{=}7{,}45$

10 Resultados

O principal objetivo desse trabalho é avaliar o desempenho da AMG frente a outros algoritmos conhecidos pela literatura. O algoritmo foi desenvolvido pelos autores deste artigo no software R e está disponível em:https://github.com/AlbertoRodrigues/generalized-model-tree-code. Nos resultados foram testados diversos algoritmos consolidados na literatura e assim comparamos as performances desses algoritmos com a AMG. Para avaliação da performance inicialmente foram separados 30% para o conjunto de teste, que será utilizado para comparação entre os algoritmos e 70% para o conjunto de treino em que será utilizado para otimização de hiperparâmetros utilizando 5 validações cruzadas de 3 folds com uma busca aleatória de 30 iterações. Em todos os algoritmos testados utilizamos o erro médio absoluto. Os algoritmos utilizados foram:

- Árvore de Modelos Lineares Generalizados;
- Regressão *Ridge*;
- Regressão Lasso;
- Floresta Aleatória;
- Árvore de Regressão;
- *Gradiente Tree Boosting*;
- Modelo linear generalizado com distribuição Normal;
- Modelo linear generalizado com distribuição Gama;
- Modelo linear generalizado com distribuição Normal Inversa.

10.1 Pré-processamento de dados

O pré-processamento básico foi feito excluindo observações com dados faltantes, que ocorreram em poucas ocasiões e realizando seleção de variáveis. O método utilizado para seleção de variáveis é abordado a seguir. Considere V o conjunto de variáveis do conjunto de dados.

1. $\forall\ V_k \in V$ é calculada a correlação linear de Pearson.
2. É obtido um ranking R de variáveis ordenadas com maior correlação em módulo.
3. É ajustado um modelo de *Gradient Tree Boosting* com o acréscimo da variável R_i, $\forall i = 1, ..., \#R$.
4. São calculados os erros de predição para cada modelo do item anterior e são selecionadas as variáveis que obtiveram um bom erro de predição com um subconjunto das variáveis.

10.2 Aplicações em conjunto de dados reais

Os dados consistem de 50 unidades amostrais de três espécies (setosa, virginica, versicolor). Para esse estudo a previsão foi realizada com base na variável resposta comprimento da sépala com base nas variáveis explicativas: largura da sépala, comprimento da pétala e largura da pétala.

Informações sobre o conjuntos de dados Iris

Quantidade de variáveis	Quantidade de observações
5	150

Resultados dos algoritmos em termos de erro médio absoluto

Algoritmos	Erro
AMG	0,2596
MLG Normal	0,3248
MLG Gama	0,3222
MLG Normal Inversa	0,4567
Ridge	0,2725
Lasso	0,3937
Floresta Aleatória	0,2959
Árvore de Regressão	0,3294
Gradient Tree	0,2870

Os dados são a respeito de consumo de combustível de carros em milhas por galão. Nesse estudo prevemos a variável resposta aceleração com base nas variáveis explicativas: peso, ano do modelo, quantidade de cilindros, cavalo-vapor, origem e deslocamento.

Informações sobre o conjuntos de dados Auto MPG

Quantidade de variáveis	Quantidade de observações
8	398

Resultados dos algoritmos em termos de erro médio absoluto

Algoritmos	Erro
AMG	1,1809
MLG Normal	1,3781
MLG Gama	2,1743
MLG Normal Inversa	2,1743
Ridge	1,2231
Lasso	1,2235
Floresta Aleatória	1,2633
Árvore de Regressão	1,3732
Gradient Tree	1,2291

Nesse estudo, o interesse é prever a resistência da compressão do concreto. As variáveis do conjunto de dados são: cimento, água, resistência a compressão de 28 dias e algumas outras relacionadas ao material.

Informações sobre o conjuntos de dados Concrete Slump Test

Quantidade de variáveis	Quantidade de observações
10	104

Resultados dos algoritmos em termos de erro médio absoluto

Algoritmos	Erro
AMG	10,1334
MLG Normal	2,1047
MLG Gama	2,8046
MLG Normal Inversa	5,7530
Ridge	2,1718
Lasso	2,1639
Floresta Aleatória	2,6980
Árvore de Regressão	3,2827
Gradient Tree	2,0414

O interesse desse estudo é prever o valor das casas com base nas variáveis explicativas: idade da casa, distância da estação mais próxima, número de lojas de conveniência em uma área circular, latitude e longitude.

Informações sobre o conjuntos de dados Real estate valuation

Quantidade de variáveis	Quantidade de observações
8	414

Resultados dos algoritmos em termos de erro médio absoluto

Algoritmos	Erro
AMG	6,7526
MLG Normal	6,2092
MLG Gama	10,4060
MLG Normal Inversa	10,4060
Ridge	7,2513
Lasso	7,3225
Floresta Aleatória	5,9433
Árvore de Regressão	6,6193
Gradient Tree	5,9128

Este conjunto de dados é composto por observações de pessoas com doença de Parkinson em estágio inicial com o objetivo de prever a pontuações motoras com base em medidas de voz.

Informações sobre o conjuntos de dados Parkinsons

Quantidade de variáveis	Quantidade de observações
22	5875

Resultados dos algoritmos em termos de erro médio absoluto

Algoritmos	Erro
AMG	5,5597
MLG Normal	2,0366
MLG Gama	2,6938
MLG Normal Inversa	2,6858
Ridge	1,9302
Lasso	1,9316
Floresta Aleatória	1,3513
Árvore de Regressão	1,5050
Gradient Tree	0,3124

Este conjunto de dados tem como objetivo prever um composto molecular com base em propriedades moleculares para ajudar a entender a toxicidade aquática aguda para os peixes Pimephales promelas, que causa morte.

Informações sobre o conjuntos de dados qsar toxicity

Quantidade de variáveis	Quantidade de observações
7	908

Resultados dos algoritmos em termos de erro médio absoluto

Algoritmos	Erro
AMG	0,2484
MLG Normal	0,2988
MLG Gama	0,2973
MLG Normal Inversa	0,2997
Ridge	0,2657
Lasso	0,2967
Floresta Aleatória	0,2424
Árvore de Regressão	0,2571
Gradient Tree	0,2273

Foram realizadas análises de energia usando 12 diferentes formatos de edifícios simulados no Ecotect. Os edifícios diferem no que diz respeito à área envidraçada, à distribuição da área envidraçada, à orientação, entre outros parâmetros.O objetivo principal é prever carga de resfriamento com base em compacidade relativa, área de Superfície, área da parede, área do telhado, altura geral, orientação, área de envidraçamento e distribuição da área de envidraçamento.

Informações sobre o conjuntos de dados energy efficiency

Quantidade de variáveis	Quantidade de observações
10	768

Resultados dos algoritmos em termos de erro médio absoluto

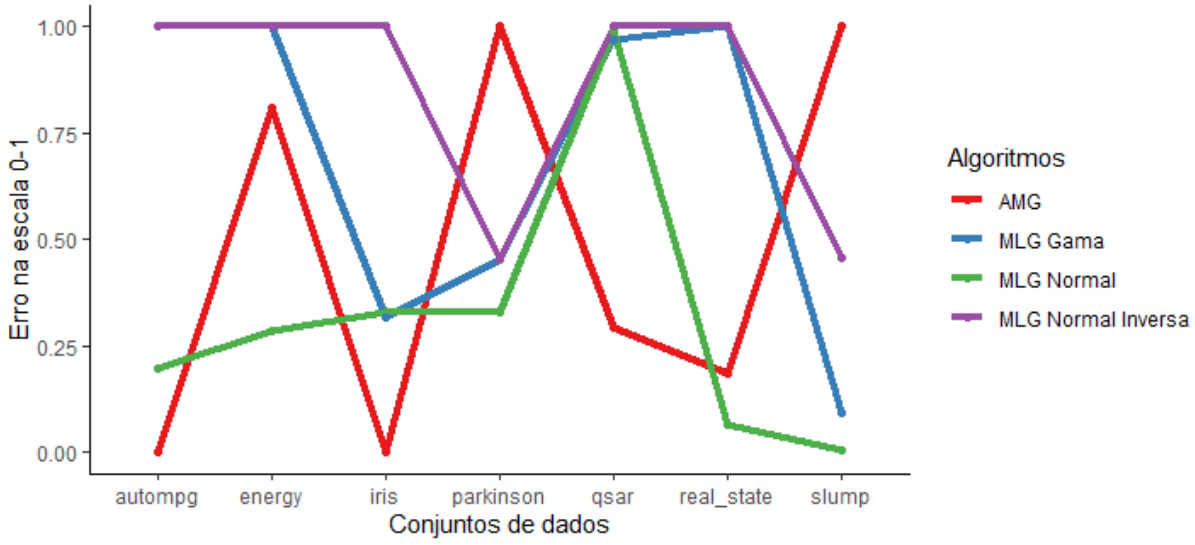
Algoritmos	Erro
AMG	7,2601
MLG Normal	3,2152
MLG Gama	8,7412
MLG Normal Inversa	8,7412
Ridge	2,3229
Lasso	2,5002
Floresta Aleatória	1,3767
Árvore de Regressão	1,4341
Gradient Tree	1,0139

10.3 Comparações dos erros

Neste tópico, serão apresentados gráficos de comparação entre nosso modelo proposto e três tipos de classe de modelos: Modelos lineares generalizados, Modelos lineares com regularização e Modelos baseados em árvore. Serão mostrados somente os erros de cada algoritmo em todos os conjuntos de dados apresentado em 10.2, os erros estão redimensionados para a escala [0,1] para uma melhor visualização.

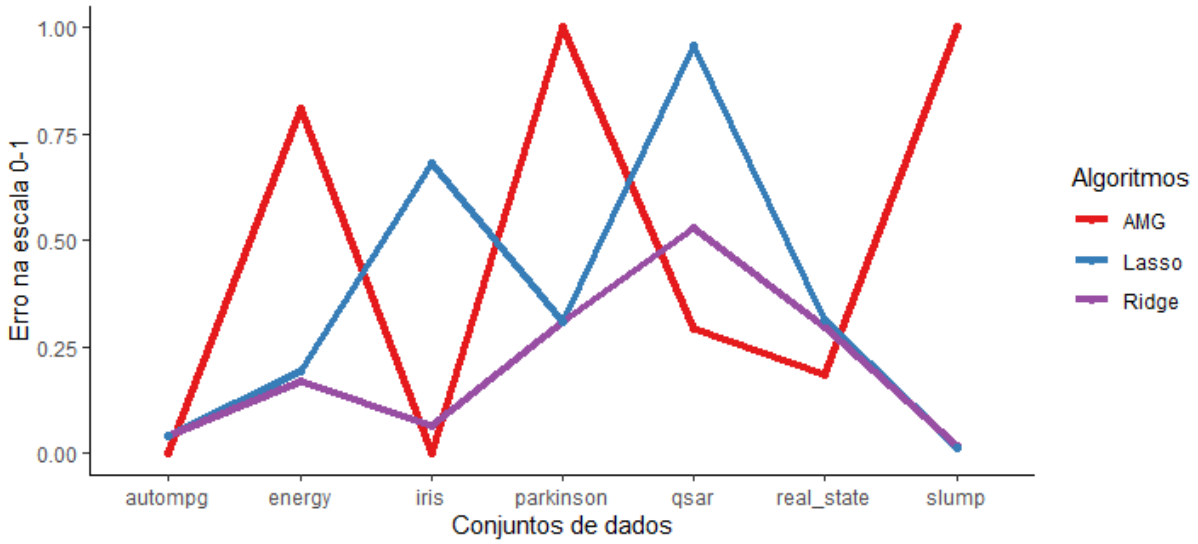
Na figura 2 percebemos que em alguns dos conjuntos de dados a AMG obtém um erro menor que os MLGs e em outras situações obteve o segundo menor erro. Assim, obtendo dos 7 conjunto de dados apresentados 5 obtendo o primeiro ou segundo menor erro de teste. Isso de fato é esperado, pois além dos MLGs serem um caso particular da AMG, a ideia desse artigo é melhorar a predição da variável resposta do que um único MLG.

Figura 2: Comparações entre os MLGs



Na figura 3, novamente podemos notar que maioria das vezes a AMG obteve um erro menor que os modelos lineares com regularização.

Figura 3: Comparação entre os modelos lineares com regularização



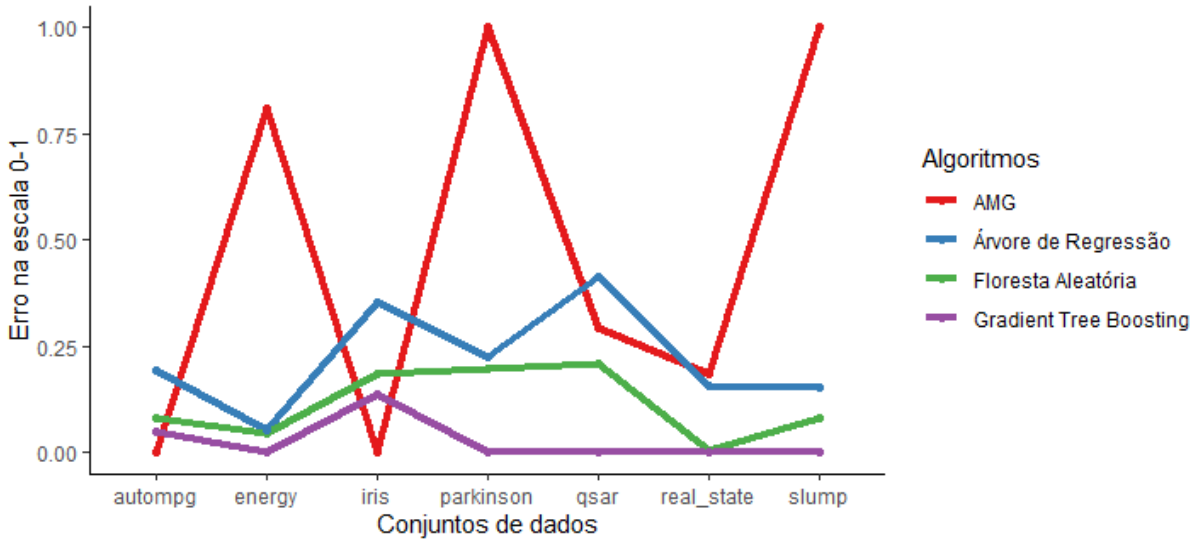
Na figura 4 percebemos uma maior competitividade entre os algoritmos baseados em árvores. Nesse caso, notamos que na maioria dos conjunto de dados o erro do algoritmo *Gradient Boosting Tree* é o menor, que é um algoritmo muito conhecido por sua robustez e também porque na prática costuma atingir as melhores performances em competições de *Machine Learning* como na plataforma Kaggle, porém no conjunto Iris e Autompg, a AMG obteve um melhor desempenho.

Na tabela 1, é mostrada a quantidade e a proporção de conjuntos de dados(de um total de 7) em que o erro do conjunto de teste AMG é menor que o erro do conjunto de teste de determinado algoritmo, observado no tópico 10.2. Esta é uma maneira de verificarmos que a AMG de fato se destacou aos vários conjuntos de dados apresentados, de diferentes tamanhos e quantidade de variáveis.

Tabela 1: Comparação entre os erros no conjunto de teste

Algoritmos	Quantidade	Proporção
MLG Normal	3	0,4286
MLG Gama	5	0,7143
MLG Normal Inversa	5	0,7143
Regressão Ridge	4	0.5714
Regressão Lasso	4	0.5714
Floresta Aleatória	2	0.2857
Árvore de Regressão	3	0.4286
Gradient Tree Boosting	2	0.2857

Figura 4: Comparação entre os modelos baseados em árvore



11 Conclusão

Neste artigo, vimos um alternativa flexível de algoritmo para predição de uma variável resposta contínua em que sua construção é baseada no particionamento do conjunto de dados com o intuito de minimizar a variação da variável resposta.

Na seção 10.2, obtivemos uma performance satisfatória em muitos conjuntos de dados, competindo muito bem com algoritmos bem consolidados na literatura. Em alguns conjuntos de dados obtivemos a melhor performance dentre todos os algoritmos testados, como no conjunto de dados iris e slump.

Na comparação individual contra os MLGs, presente na figura 2, foi mostrado que sempre o erro de teste da AMG é obteve melhores resultados comparativamente com os MLGs, mostrando sua eficácia contra uma classe de modelos estatísticos bastante conhecida. Com 71,43% dos conjunto de dados tendo melhores desempenhos que os modelos lineares generalizados Gama e Normal e Inversa. Desta forma, em conjunto de dados reais a AMG é um bom concorrente contra essa classe de modelos.

Na comparação contra a classe modelos lineares com regularização obtivemos uma performance superior do modelo Lasso e Ridge em 57,7% dos conjuntos de dados, indicando uma superioridade em predição com esses conjunto de dados.

Na comparação contra a classe de modelos baseados em árvore, ocorreu uma maior competitividade entre os algoritmos. Principalmente nos algoritmos Floresta aleatória e *Gradient Tree Boosting* existe a vantagem de redução da variância do modelo pelo motivo de se combinar muitas árvores e ter uma amostragem de variáveis, essa redução de variância ocasiona em várias situações uma melhora significativa na predição da variável resposta e pelo fato que a AMG ser uma única árvore, pode ser que haja uma variância maior e que prejudique

a predição. Em 28,57% dos conjunto de dados a AMG se mostrou ser superior do que os algoritmos Floresta Aleatória e *Gradient Tree Boosting* e em apenas 42,86% teve desempenho superior contra o algoritmo Árvore de Regressão.

Vimos que a AMG possui uma ampla variedade de modelos para serem ajustados aos dados e que contemplam mais opções de predição para a variável resposta. A modelagem baseada na AMG em dados reais permite que profissionais possam usar esse modelo na prática para visualização, interpretação e predição de uma variável contínua de forma simples e robusta.

Referências

1. Akaike, H.: A new look at the statistical model identification. IEEE transactions on automatic control **19**(6), 716–723 (1974)
2. Alexander, W.P., Grimshaw, S.D.: Treed regression. Journal of Computational and Graphical Statistics **5**(2), 156–175 (1996)
3. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and regression trees (1984)
4. Chaudhuri, P., Huang, M.C., Loh, W.Y., Yao, R.: Piecewise-polynomial regression trees. Statistica Sinica pp. 143–167 (1994)
5. Chaudhuri, P., Lo, W.D., Loh, W.Y., Yang, C.C.: Generalized regression trees. Statistica Sinica pp. 641–666 (1995)
6. Chipman, H.A., George, E.I., McCulloch, R.E.: Bayesian treed models. Machine Learning **48**(1-3), 299–320 (2002)
7. Cordeiro, G.M., Demétrio, C.G.: Modelos lineares generalizados e extensões (2007)
8. Frank, E., Wang, Y., Inglis, S., Holmes, G., Witten, I.H.: Using model trees for classification. Machine learning **32**(1), 63–76 (1998)
9. Landwehr, N., Hall, M., Frank, E.: Logistic model trees. Machine learning **59**(1-2), 161–205 (2005)
10. Paula, G.A.: Modelos de regressão com apoio computacional
11. Quinlan, J.R.: C4. 5: programs for machine learning (1993)
12. Quinlan, J.R., et al.: Learning with continuous classes. In: 5th Australian joint conference on artificial intelligence. vol. 92, pp. 343–348. World Scientific (1992)
13. Wang, Y., Witten, I.H.: Induction of model trees for predicting continuous classes (1996)
14. Zeileis, A., Hothorn, T., Hornik, K.: Model-based recursive partitioning. Journal of Computational and Graphical Statistics **17**(2), 492–514 (2008)