Seminário I — Algoritmo — Mineração de Dados

Implementação do *Kernel SHAP*

Guilherme Seidyo Imai Aldeia

Disciplina Mineração de dados, ministrada pelo Prof. Dr. Thiago Covões
Programa de Pós Graduação em Ciência da Computação
Universidade Federal do ABC (UFABC)

Índice

Artigo selecionado

- A Unified Approach to Interpreting Model Predictions
- Scott M. Lundberg, Su-In Lee
- Publicado em 2017
- 1918 citações!

A Unified Approach to Interpreting Model Predictions

Scott M. Lundberg
Paul G. Allen School of Computer Science
University of Washington
Seattle, WA 98105
slund1@cs.washington.edu

Su-In Lee
Paul G. Allen School of Computer Science
Department of Genome Sciences
University of Washington
Seattle, WA 98105
suinlee@cs.washington.edu

Abstract

Understanding why a model makes a certain prediction can be as crucial as the prediction's accuracy in many applications. However, the highest accuracy for large modern datasets is often achieved by complex models that even experts struggle to interpret, such as ensemble or deep learning models, creating a tension between *accuracy* and *interpretability*. In response, various methods have recently been proposed to help users interpret the predictions of complex models, but it is often unclear how these methods are related and when one method is preferable over another. To address this problem, we present a unified framework for interpreting predictions, SHAP (SHapley Additive exPlanations). SHAP assigns each feature an importance value for a particular prediction. Its novel components include: (1) the identification of a new class of additive feature importance measures, and (2) theoretical results showing there is a unique solution in this class with a set of desirable properties. The new class unifies six existing methods, notable because several recent methods in the class lack the proposed desirable properties. Based on insights from this unification, we present new methods that show improved computational performance and/or better consistency with human intuition than previous approaches.

Overview

- Identifica uma nova forma para entender as predições que um modelo faz:
  - Para interpretar um modelo usamos outro modelo (modelo de explicação)
- Propõe um *framework* para interpretar predições:
  - **SHAP** (**SH**apley **A**dditive ex**P**lanation)
  - Agnóstico de modelo
  - Unificação de 6 diferentes modelos da literatura
    - *LIME*, *DeepLIFT*, *Layer-Wise Relevance Propagation*, *Shapley Regression Values*, *Shapley Sampling Values*, *Quantitative Input Influence*

Índice

LIME

Seja f o modelo de predição e g o modelo de explicação, primeiro simplifica a entrada x' com um mapeamento $x = h_x(x')$, convertendo um vetor binário de entradas ao espaço original. Assumindo:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i,$$

onde $z' \in \{0,1\}^M$, e M é o número de atributos simplificados. Então, busca minimizar a expressão:

$$\xi = \arg\min_{g \in G} L(f, g, \pi_{x'}) + \Omega(g),$$

onde L é o erro quadrático, Ω é uma penalidade da complexidade de g, e $\pi_{x'}$ é um *kernel* de ponderação.

LIME

A solução é um efeito ϕ_i para cada atributo, sendo que a soma do efeito de todos os atributos deve aproximar a saída do modelo original.

Shapley Regression Values

Calcula a importância de atributos para modelos lineares quando há multicolinearidade.
Treina o modelo para todos os subconjuntos de atributos $S \subseteq F$, onde F é o conjunto de todos os atributos, e então para cada atributo, são computados a diferença de performance do modelo com e sem ele:

$$f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S),$$

onde x_S representa os valores dos atributos no conjunto S.
Uma vez que o valor de remover um atributo depende dos outros atributos do modelo, isso é computado para todo possível subconjunto $S \subseteq F \setminus \{i\}$.

Shapley Regression Values

Os valores *Shapley* são computados e usados como importância dos atributos, à partir de uma média ponderada de todas as possíveis diferenças:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)].$$

Nesse caso, h_x mapeia 1 ou 0 para o espaço original de entrada, com 1 indicando a presença e 0 indicando a exclusão do atributo no modelo.

Propriedades

Embora diferentes, esses métodos fazem parte de uma mesma classe, contanto que atendam às propriedades:

- **Acurácia local.** Ao aproximar um modelo original f para uma entrada x, o modelo de explicação deve ter o mesmo valor de f para a entrada simplificada x' $f(x) = g(x')$;
- **Ausência de valores.** Se as entradas simplificadas representam a ausência de atributos, deve valer que a ausência de atributos não tem impacto no modelo original: $x'_i = 0 \Rightarrow \phi_i = 0.$;
- **Consistência.** Se um modelo muda a contribuição de um atributo (aumenta ou se mantém), indiferente das entradas, a importância atribuída não deve diminuir.

Teorema 1 (Valores *Shapley*)

Apenas uma possível explicação $g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i$ satisfaz a propriedades 1-3:

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!}[f_x(z') - f_x(z' \setminus i)],$$

onde $|z'|$ é o número de atributos diferentes de zero em z', e $z' \subseteq x'$ representa todos os vetores z' onde os atributos diferentes de zero são um subconjunto dos atributos de x' diferentes de zero.

Implicações do Teorema 1

- Possível definir o SHAP como uma medida unificada de importância de atributos;
- As importâncias (valores SHAP, ou *Shapley*) são obtidas com a solução da equação do teorema.

Teorema 2 (*Kernel SHAP*)

Os valores que tornam o LIME consistente (Propriedade 3) são:

$$\Omega(g) = 0,$$

$$\pi_{x'}(z') = \frac{M-1}{\left(\binom{M}{|z'|}\right)|z'|(M-|z'|)},$$

$$L(f, g, \pi_{x'}) = \sum_{z' \in Z} [f(h_x(z')) - g(z')]^2 \pi_{x'}(z'),$$

onde $|z'|$ é o número de elementos diferentes de zero em z'.

Implicações do Teorema 2

- O LIME supõe $g(z')$ é uma suposição de que possui uma forma linhar, e L é a função de custo quadrática, de forma que a equação do LIME ainda possa ser resolvida com uma regressão linear;
- Isso leva a um *kernel* que difere nitidamente dos núcleos escolhidos heuristicamente anteriores (no artigo do LIME);
- Sabemos que um ponto que minimiza o erro médio quadrático de uma série de dados é a média dessa série, portanto existe uma forma de fazer com que a regressão linear encontre os valores *Shapley*!

Índice

Criação de um pseudo-código

- Artigo original não tem pseudo-código
- Criei um para ilustrar o algoritmo!
  - Minha contribuição
  - Inspirado no código em Python dos autores
  - Possibilita melhor entendimento dos cálculos feitos pelo algoritmo

Algoritmo Kernel SHAP

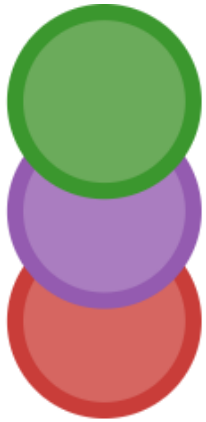
Algoritmo 1: Algoritmo Kernel SHAP

Entrada: Modelo de aprendizado f, dados para explicar X, número de atributos M, referência r
Saída : Contribuição das variáveis ϕ (valor base dado por ϕ_M)

```
1  indicesAtributos ← ∅;
2  // Gerando todas as possíveis combinações de presenças e ausências dos atributos
3  // (Note que a ausência de todos os atributos (0) faz parte dos casos)
4  para i ∈ {0, 1, 2, ... M} faça
5      // geraCombinacoes(xs, M) gera todas as combinações de tamanho M dos elementos em xs
6      indicesAtributos ← indicesAtributos ∪ geraCombinacoes(i);
7  X ← (0_(2^M, M) | 1_(2^M, 1));
8  // Matriz onde cada linha será um caso onde há variáveis ausentes
9  V ← 0_(2^M, M);
10 pesos ← ∅;
11 para (i, s) ∈ enumerar(indicesAtributos) faça
12     // Fixando variáveis na referência e alterando apenas as que são consideradas
13     V [i ] ← r;
14     V [i,s ] ← X [s ];
15     // Matriz onde cada linha contém 1 indicando presença e 0 indicando ausência
16     X [i, s ] ← 1;
17     // Equação do Teorema 2
18     pesos [i] ← kernelShapley(M, |s|);
19 // Aplicando a função para cada caso possível
20 y ← f(V);
21 // Método dos mínimos quadrados com pesos, onde diagonal(pesos) denota uma
22 // matriz diagonal com os elementos em pesos
23 retorna (X^T · diagonal(pesos) · X)^−1 (X^T · diagonal(pesos) · y);
```

Índice

Demonstração

Pluto.jl 

Índice

Conclusões

- Projeto ajudou a consolidar conceitos vistos no curso
- O estudo feito contribuiu para meu projeto de mestrado
- Algoritmo implementado fornece explicações locais para predições feitas por um modelo de aprendizado de máquina
- Explicação global pode ser obtida pelas explicações locais
- Implementação é simples, mas o entendimento requer grande bagagem teórica
- Julia e Pluto são legais

Dificuldades

Teóricas

- Entendimento requer conhecimentos matemáticos (regressão, MMQ, e entendimento das fórmulas).

Implementação

- Ausência de pseudo-código;
- Linguagem Julia.

Questões para reflexão

- Aproximar um modelo complexo com um modelo simples é suficiente?
- As limitações do modelo de explicação restringem o poder de explicação?
- Um modelo de explicação pode ser caixa preta?
- A melhor explicação para um modelo não deveria ser o próprio modelo?

Obrigado!

Seminário I — Algoritmo — Mineração de dados:
Implementação do *Kernel SHAP*

Guilherme Seidyo Imai Aldeia (guilherme.aldeia@ufabc.edu.br)