Regresyjny las losowy

Eryk Warchulski

1 Wprowadzenie.

Tematem projektu jest implementacja regresyjnego lasu losowego w zmodyfikowanej wersji w języku programowania **R**. W punkcie 2 zostanie rozwinięty temat projektu, tj. zaprezentowanie idei stojącej za regresyjnym lasem losowym. Zostanie również krótko opisana modyfikacja algorytmu, której dokładniejszy opis znajdzie się w punkcie 3. W tym też punkcie zostanie przedstawione podłoże teoretyczne, na której opiera się modyfikacja oraz przykłady obliczeniowe. Następnie, tj. w punkcie 4, przedstawię planowane eksperymenty obliczeniowe oraz ich analizę statystyczną. Punkt 5 będzie podsumowaniem, w którym będą pokrótce opisane sprawy związane z implementacją programową.

2 Regresyjny las losowy.

Przed przystąpieniem do opisania lasu regresyjnego warto byłoby najpierw zacząć od przedstawienia jego składowych, tj. drzew regresyjnych.
Drzewem regresyjnym nazywa się drzewo decyzyjne, które na podstawie danego zbioru uczącego $\mathcal{T}$ służy do rozwiązania zadania regresji. Zadaniem regresji nazywa się problem znalezienia przekształcenia odwzorowania p zmiennych $\mathbf{x} \in \mathbb{R}^p$ w $y \in \mathbb{R}$, które minimalizuje pewną funkcję straty $\mathcal{L}$. Zmienne $\mathbf{x} = \big(x1, x2, ..., x_p\big)$ nazywa się *zmiennymi objaśniającymi*, a zmienną y *zmienną objaśnianą*.
Niech f będzie rzeczywistym odwzorowaniem zmiennych objaśniających na zmienną objaśnianą:

$$f \colon \mathbb{R}^p \to \mathbb{R} \tag{1}$$

Zadanie regresji sprowadza się do znalezienia takiego $\hat{f}$, że:

$$\hat{f}^* \leftarrow \underset{\hat{f} \in \mathcal{F}}{\operatorname{argmin}} \mathcal{L}(f, \hat{f}) \tag{2}$$

gdzie $\mathcal{F}$ jest przestrzenią funkcyjną z określoną normą. Wybór funkcji straty jest bardzo istotnym zagadnieniem w problemie regresji i najczęściej pada on na MISE:

$$\mathcal{L}(f, \hat{f}) \leftarrow \mathbb{E} \left\| f - \hat{f} \right\|_2^2 \tag{3}$$

W rzeczywistości dysponujemy wyłącznie N próbkami w postaci par $(\mathbf{x}_i, y_i)$, więc empirycznym odpowiednikiem powyższej funkcji straty będzie

$$\hat{\mathcal{L}}(f, \hat{f}; N) \leftarrow \frac{1}{N} \sum_{i=1}^{N} \big[y_i - \hat{f}(x_i) \big]^2 \tag{4}$$

Drzewa regresyjne są metodą regresji nieparametrycznej, co oznacza, że nie czynią żadnych wstępnych założeń odnośnie modelowanej funkcji f. W metodach parametrycznych zakłada się, że funkcja f jest zależna od θ, będącego w ogólności wektorem parametrów i celem jej jest znalezienie takich wartości parametrów, aby zminimalizować funkcję straty.

Zwracany model z drzewa regresyjnego można przedstawić jako:

$$\hat{f}(\mathbf{x}) = \sum_{m=1}^{\#\mathcal{M}} c_m \mathbb{1}(\mathbf{x}_m \in R_m) \tag{5}$$

co jest równoważne temu, że drzewo dzieli przestrzeń na M p-wymiarowych hiper-prostopadłościanów, które mają przybliżać poszukiwaną funkcję f.
Istnieje kilka popularnych metod konstrukcji drzewa regresyjnego jak AID, C4.5. W projekcie zostaną wykorzystane drzewa CART opisane przez L. Breimana, który jest również twórcą metody lasu regresyjnego. W metodzie tej do konstrukcji drzewa decyzyjnego stosuje się binarny podział rekurencyjny danych ze zbioru uczącego $\mathcal{T}$. Algorytm musi znaleźć optymalną zmienną dzielącą oraz punkt podziału. Po jego znalezieniu dane są dzielone na dwa zbiory w zależności od tego, czy są większe czy mniejsze od punktu podziału dla danej zmiennej dzielącej. Proces ten jest powtarzany na obu wynikach podziału, aż do osiągnięcia maksymalnej głębokości drzewa określonej przez minimalną pojemność węzła, tj. liczbę przykładów, które do niego trafią. Im mniejsza jest pojemność węzła, tym drzewo jest głębsze. Liście tego drzewa wyznaczają regiony R_m.
Okazuje się, że drzewa regresyjne są bardzo wrażliwe na zmiany w zbiorze uczącym i aby temu zapobiec próbuje się t.zw. metody *baggingu* drzew. Polega on agregacji drzew konstruowanych na podstawie próby bootstrapowej ze zbioru uczącego $\mathcal{T}$. Konstruuje się B drzew na zbiorach boostrapowych $\mathcal{T}^*$, a ostateczny wynik drzewa – w przypadku regresji – jest średnią ze wszystkich B drzew:

$$\hat{f}(\mathbf{x}) = \frac{1}{B} \sum_{i=1}^{B} \hat{f}(\mathbf{x})_i \tag{6}$$

Warto jednak pokusić się o analizę rozrzutu takiej średniej.

$$\mathbb{V}\big[\hat{f}(\mathbf{x})\big] = \mathbb{V}\big[\frac{1}{B} \sum B_i \hat{f}(\mathbf{x})_i\big] = \frac{1}{B^2} \mathbb{V}\big[\sum_{i}^{B} \hat{f}(\mathbf{x})_i\big] \tag{7}$$

Jeśli zakładamy, że wszystkie zmienne są z tego samego rozkładu o wartości pierwszego i drugiego momentu wynoszą odpowiednio m, σ^2 oraz zmienne są niezależne, to powyższa równość sprowadza się do trywialnej postaci:

$$\mathbb{V}\big[\hat{f}(\mathbf{x})\big] = \frac{\sigma^2}{B^2} \tag{8}$$

Oznacza to, że wystarczyłoby odpowiednio zwiększać liczbę drzew, aby zminimalizować wariancję. W ogólności założenie niezależności zmiennych nie jest

uzasadnione i zwiększenie liczby drzew, które wiąże się z oczywistym przyrostem kosztu obliczeniowego, nie zmniejszy wariancji w tak prosty sposób.
Przy założeniu zależności zmiennych losowych wyprowadzenie zależności na wariancję będzie wymagało wprowadzenia następujących własności:

Korelacją dwóch zmiennych losowych X_i i X_j nazywa się następującą wielkość

$$\rho = \frac{1}{\sigma^2}\mathbb{E}\big[(X_i - m)(X_j - m)\big] \tag{9}$$

Powyższą równość można przekształcić do postaci:

$$\mathbb{E}\big[X_i X_j\big] = \begin{cases} \sigma^2\rho + \mu^2, & i \neq j \\ \sigma^2 + \mu^2, & i = j \end{cases} \tag{10}$$

Na mocy powyższych zależności: (pomijam indeksy górne sum w dalszych równościach)

$$\mathbb{V}\big[\sum_i^B \hat{f}(\mathbf{x})_i\big] = \mathbb{E}\big[\sum_i \hat{f}_i(\mathbf{x})\big]^2 - \big(\mathbb{E}\big[\sum_i \hat{f}_i(\mathbf{x})\big]\big)^2 \tag{11}$$

Pierwszy składnik można zapisać jako:

$$\mathbb{E}\big[\sum_i \hat{f}_i(\mathbf{x})\big]^2 = \mathbb{E}\big[\sum_{i,j} \hat{f}_i(\mathbf{x})\hat{f}_j(\mathbf{x})\big] = \star \tag{12}$$

powyższe sumy są równe

$$\star = B\mathbb{E}\big[\hat{f}_i^2(\mathbf{x})\big] + (B^2 - B)\mathbb{E}\big[\hat{f}_i(\mathbf{x})\hat{f}_j(\mathbf{x})\big] = B(\sigma^2 + m^2) + (B^2 - B)(\rho\sigma^2 + m^2)^1 \tag{13}$$

Ostatecznie szukana wariancja wynosi:

$$\frac{1}{B^2}\mathbb{V}\big[\sum_i^B \hat{f}(\mathbf{x})_i\big] = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2 \tag{14}$$

Widać więc, że zwiększanie liczby drzew może wyeliminować drugi składnik powyższej sumy. Problematyczny jest pierwszy składnik, ponieważ zależy od korelacji i nie zależy od liczby drzew.
Problem ten dotyczy w znacznie mniejszym stopniu lasów losowych. Wynika to z faktu, że lasy losowe do konstrukcji każdego drzewa losowo wybierają m zmiennych objaśniających ze wszystkich p zmiennych, co skutkuje znaczną dekorelacją.

Na koniec warto zamieścić pseudokod, opisujący las losowy, którego implementacja jest tematem projektu:

[1] B składników o tych samych indeksach i $B^2 - B$ składników o indeksach różnych.

1: $Input : \mathcal{T}, MaxNodeSize, Ker$
2: $Konstrukcja\ lasu$:
3: **for** $i \in \{1, ..., B\}$ **do**
4: $\quad m \leftarrow rand()$
5: $\quad \hat{f}_i \leftarrow$ MakeTree(i, $\mathcal{T}$, MaxNodeSize, Ker, m)
6: **end for**
7: Działanie lasu: $\hat{f}(\mathbf{x}) \leftarrow \frac{1}{B} \sum_{i=1}^{B} \hat{f}_i(x)$

Istotna jest procedura $MakeTree()$, w której zawarta jest modyfikacja algorytmu, polegająca na tym, że kolejne drzewa są generowane na podstawie przykładów ze zbioru uczącego, na których model dawał dotychczas większy błąd.

3 Modyfikacja algorytmu.

Jak zostało wspomniane w punkcie poprzednim: modyfikacja algorytmu polega na tym, aby kolejne drzewa w budowanym lesie losowym były konstruowane na podstawie przykładów ze zbioru uczącego, na których wcześniejsze drzewa dawały większy błąd. W klasycznej wersji algorytmu kolejne drzewa są konstruowane na podstawie próby boostrapowej o rozmiarze $\mathcal{N}$, w której każdy przykład ma takie samo prawdopodobieństwo bycia wylosowanym. Niech $\mathcal{I}$ będzie zmienną losową, która oznacza indeks wylosowanego przykładu ze zbioru uczącego. W związku z tym, co zostało opisane wyżej – klasyczna wersja utrzymuje jeden rozkład prawdopodobieństwa opisujący zmienną losową $\mathcal{I}$ do generowania pseudopróby tj.:

$$\mathcal{I} \sim \mathcal{U}_{[1:\ \#\mathcal{T}]} \tag{15}$$

Należy jednak zaznaczyć, że ze względu na dyskretny charakter problemu wyboru indeksu z przeliczalnego zbioru indeksów $\mathcal{T}$ realizacja zmiennej losowej $\mathcal{I}$ jest mapowana na liczbę ze zbioru $\mathbb{N}$ za pomocą funkcji *podłogi*:

$$i \leftarrow \lfloor i \rfloor \tag{16}$$

Sprowadzenie problemu do przypadku ciągłego jest podyktowane przez fakt stosowanej metody realizującej modyfikację konstrukcji lasu losowego.
Jeśli kolejne drzewa mają być konstruowane na podstawie przykładów, które dotychczas dawały większy błąd, to należałoby doprowadzić do sytuacji, w której generator indeksów $\mathbb{G}$, będący dwójką $\langle \mathcal{T}, f_{\mathcal{I}} \rangle$, będzie aktualizował rozkład prawdopodobieństwa $f_{\mathcal{I}}$, z którym generuje indeksy. Nowy rozkład powinien przypisywać większe prawdopodobieństwo indeksom, na których dotychczasowy model popełniał większe błędy.
Realizacja powyżej opisanego mechanizmu będzie oparta o KDE, tj. *kernel density estimation* – nieparametryczną metodę estymacji funkcji gęstości prawdopodobieństwa $\hat{f}(x)$ rozkładu na podstawie danej próby losowej.
Niech $\mathbf{E} = \left[E_1, E_2, ..., E_{\#\mathcal{T}}\right]$ oznacza wektor zmiennych zliczających przykłady,

które dawały błąd regresji, oraz niech k-temu przykładowi w zbiorze trenującym odpowiada k-ta pozycja w wektorze $\mathbf{E}$:

$$E_k \leftarrow \sum_{b}^{\breve{B}} \mathbb{1}(T^b(x_k) > \epsilon) \tag{17}$$

co jest liczbą błędów popełnionych przez dotychczasowy model złożony z $\breve{B}$ drzew dla przykładu x_k. Liczby częstości wystąpień błędu dla danego przykładu mogą posłużyć do wykreślenie histogramu, który jest graficzną formą przedstawienia rozkładu empirycznego próby. Wykres taki zostanie narysowany podczas analizy przykładu obliczeniowego, a nań nałożony będzie gładki estymator gęstości jądrowej KDE wyznaczony na próbie z przykładu.
Jądrowy estymator funkcji gęstości prawdopodobieństwa budowany jest dla danej próby statystycznej $(X_1, X_2, ..., X_T)$ jako następująca średnia arytmetyczna funkcji jądrowej $K\colon \mathbb{R}^n \to \big[0, \infty\big)$[2]:

$$\hat{f}(x) = \frac{\sum_{i=1}^{T} K(x - X_i)}{Th^T} \tag{18}$$

Przy czym h jest t.zw. współczynnikiem gładkości. Parametr ten w przeciwieństwie do wyboru funkcji jądrowej istotnie wpływa na jakość estymatora KDE.
Jakość estymatora KDE może być mierzona w jako $L = \mathbb{E}\big[f(x) - \hat{f}(x)\big]^2$. Do wyznaczenia optymalnej wartości współczynnika wygładzania h^* teoretycznie można zróżniczkować L po h i przyrównując do 0 otrzymać:

$$h^* = \frac{\int K^2(x)dx}{T \int x^2 K(x)dx \int (f^{(2)}(x))^2 dx} \tag{19}$$

W celu uniknięcia całkowania oraz różniczkowania numerycznego parametr h jest najczęściej wybierany metodą *cross validation*. Ponadto, powyższa wartość optymalna h^* jest wyprowadzona przy założeniu, że $\hat{f}$ zbiega według miary do f, gdy rozmiar próby zbiega do ∞. Nie wiemy jednak jakie jest tempo tej zbieżności wraz z przyrostem liczby przykładów n. Istnieje szeroka klasa funkcji jądrowych, tj. funkcji spełniających następujące warunki:

1. $\int_{\mathbb{R}^n} K(\mathbf{x})d\mathbf{x} = 1$
2. $(\forall \mathbf{x} \in \mathbb{R}^n)\colon K(\mathbf{0}) \geqslant K(\mathbf{x})$
3. $(\forall \mathbf{x} \in \mathbb{R}^n)\colon K(\mathbf{x}) = K(-\mathbf{x})$

Powyższe warunki zostały podane dla przypadku, gdy: $n \geqslant 1$.
Popularnymi funkcjami jądrowymi są na przykład:

[2] Funkcja ta nazywana jest również *funkcją wygładzającą*.

1. jądro Epanecznikowa:

$$K(x) \leftarrow \frac{3}{4}(1 - \|x\|^2) * \mathbb{1}\big[\|x\| \leqslant 1\big] + 0 * \mathbb{1}\big[\|x\| > 1\big] \tag{20}$$

2. jądro gaussowskie:

$$K(x) \leftarrow (2\pi)^{-\frac{1}{2}} e^{-\frac{x^2}{2}} \tag{21}$$

Ponadto wyróżnia się jeszcze jądra: jednostajne, gamma, dwuwagowe.

Po wprowadzeniu do estymacji funkcji gęstości prawdopodobieństwa za pomocą funkcji jądrowych można przejść do przykładu obliczeniowego. Będzie on prezentował idee stojącą za modyfikacją algorytmu w działaniu.

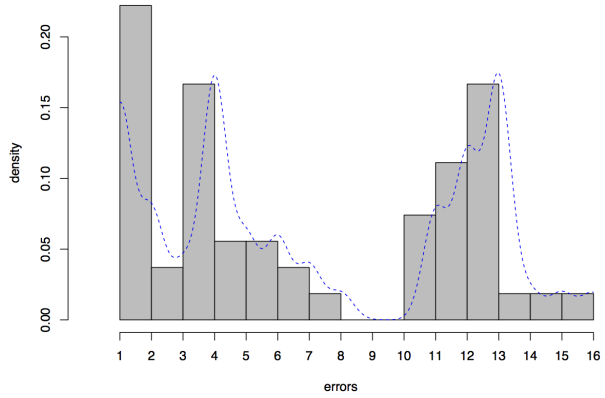
Niech $\mathcal{T} \leftarrow \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T\}$ oznacza zbiór przykładów uczących, w którym każdy element $\mathbf{x}$ jest p-wymiarowym wektorem liczb rzeczywistych, tj.:

$$(\forall \mathbf{x} \in \mathcal{T})\mathbf{x}_k = (x_{1k}, x_{2k}, ..., x_{pk}) \tag{22}$$

Z kolei $\{T_b(\mathbf{x})\}_{b=1}^{\check{B}}$ jest zbiorem dotychczas skonstruowanych drzew regresyjnych. Dla każdego drzewa ze zbioru podawane są wszystkie przykłady ze zbioru uczącego. Następnie dla każdego przykładu badana jest różnica między wartością zwróconą przez drzewo, a faktyczną wartością zmiennej objaśnianej oraz inkrementowana – bądź nie – zmienna zliczająca błędy przykładu.
Dla ustalenia uwagi niech zbiór uczący $\mathcal{T}$ składa się z 16 przykładów, a dotychczasowy las z 15 drzew. Liczba drzew w lesie jednocześnie ogranicza maksymalne wartości zmiennych zliczających. Po podaniu wszystkich przykładów ze zbioru uczącego i sprawdzeniu powyżej opisanych różnic otrzymano następujący wektor błędów $\mathbf{E}$:

$$\mathbf{E} = \big[8, 4, 2, 9, 3, 3, 2, 1, 0, 0, 4, 6, 9, 1, 1, 1\big] \tag{23}$$

Co oznacza, że dla przykładu $\mathbf{x}_1$ naliczono 8 błędów, etc. Dla takiego wektora błędów histogram wraz z naniesionym estymatorem gęstości prawdopodobieństwa wygląda następująco:



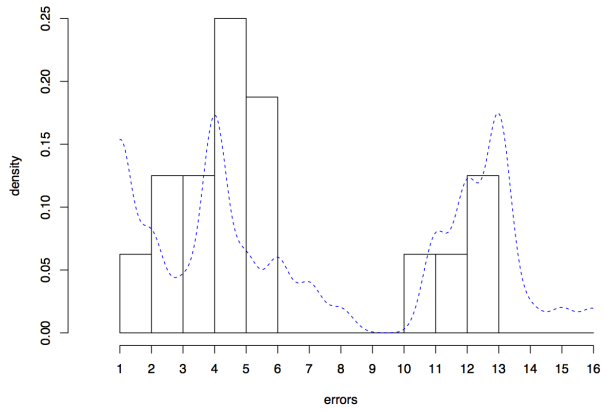
Rysunek 1.

Do wyznaczenia estymatora $\hat{f}(x)$ została użyta gaussowska funkcja jądrowa ze współczynnikiem $h = 0.392$. Użycie takiej funkcji jądrowej pozwala na stosunkowo łatwą generację programową liczb losowych z rozkładu opisanego przez $\hat{f}(x)$.
Należy zauważyć, że estymator jest sumą przesuniętych rozkładów $\mathcal{N}(0, 1)$ o wartość elementu próby losowej x_i. Każdy element sumy jest przemnożony przez czynnik h, więc odchylenie standardowe takiego rozkładu wynosi $\sigma = h$. W związku z tym należy wygenerować $\#\mathcal{T}$ liczb z rozkładu normalnego, w którym średnie będą wartościami losowanymi ze zwracaniem z próby losowej.
Implementacja takiego rozwiązania w pseudokodzie zbliżonym do języka **R** wygląda następująco:

1: means $\leftarrow$ sample(errors, $\#\mathcal{T}$, replace $= 1$)
2: numbers $\leftarrow \lfloor$(rnorm($\#\mathcal{T}, \mu =$ means, $\sigma = 0.392$))$\rfloor$

Poniżej znajduje się przykładowy wektor indeksów wygenerowanych zgodnie z rozkładem $\hat{f}(x)$:

$$\mathbf{I} = \big[13, 2, 3, 12, 2, 13, 1, 3, 3, 12, 3, 1, 6, 8, 11, 1\big] \tag{24}$$



Rysunek 2.

Analizując histogramy można stwierdzić, że wyestymowany rozkład na podstawie wektora błędów wygenerował częściej te indeksy, na których dotychczasowy model dawał większy błąd.

4 Eksperymenty obliczeniowe oraz analiza ich wyników.

Algorytm jest strojony za pomocą dwóch parametrów: liczby wybranych zmiennych objaśniających m oraz jądra Ker, które jest używane do estymacji rozkładu generującego indeksy. Daje to naturalną możliwość przeprowadzania eksperymentów obliczeniowych opartych o różne wartości m oraz Ker i porównanie otrzymanych wyników. Badany będzie błąd regresji oraz czas skonstruowania pełnego modelu. Zostanie również porównana wydajność między zmodyfikowaną wersją algorytmu, a jego klasyczną realizacją. Do porównania wyników posłużę się wykreśleniem błędu regresji w funkcji liczby drzew, wartości parametru m oraz Ker. Ponadto przeprowadzę nieparametryczny test statystyczny Wilcoxona na uzyskanych wynikach.

5 Zakończenie.

Projekt zostanie napisany w języku **R** z użyciem zewnętrznych pakietów *tidyverse*. Konstruowane drzewa będą oparte o metodę CART, którą zaimplementuję własnoręcznie, a do eksperymentów dotyczących porównania wersji modyfikowanej i klasycznej użyję pakietu *randomForest*.