No Free Lunch Theorems for Optimization

Piotr Kosakowski

Politechnika Warszawska

27.04.2023

Etymologia

"There's no such thing as a free lunch"

Wprowadzenie

No free lunch theorems to twierdzenia, które mówią o tym, że dla każdego algorytmu, lepsza wydajność w jednej klasie problemów jest równoważona przez wydajność w innej klasie.

Sformułowanie problemu

Założenia

Ograniczamy uwagę do optymalizacji kombinatorycznej, w której przestrzeń poszukiwań X , choć być może dość duża, jest skończona. Zakładamy ponadto, że przestrzeń możliwych wartości "kosztów" Y jest również skończona. Te ograniczenia są automatycznie spełnione dla algorytmów optymalizacyjnych działających na komputerach cyfrowych, gdzie zazwyczaj Y jest 32 lub 64 bitową reprezentacją liczb rzeczywistych.

Funkcja kosztu

Problem optymalizacyjny, zwany również funkcją kosztu, f jest reprezentowany jako przekształcenie $f : X \to Y$, a $F = Y^X$ oznacza przestrzeń możliwych problemów.

Oznaczenia

Próbka

Próbką rozmiaru m nazywamy czasowo uporządkowany zbiór m odrębnych odwiedzonych punktów i oznaczamy $d_m \equiv \{(d_m^x(1), d_m^y(1)), \ldots, (d_m^x(m), d_m^y(m))\}$, gdzie $d_m^x(i)$ oznacza wartość z X, zaś $d_m^y(i)$ to odpowiadający jej koszt z Y.

Przestrzeń próbek

Przestrzeń próbek rozmiaru m to $D_m = (X \times Y)^m$, $D \equiv \bigcup_{m \geq 0} D_m$.

Oznaczeń ciąg dalszy

Algorytm

Algorytm a jest reprezentowany jako przekształcenie
$a : d \in D \rightarrow \{x | x \notin d^x\}$

Miara wydajności

Miarą wydajności ozanczona jest przez $\Phi(d_m^y)$.
Na przykład szukając minimum f $\Phi(d_m^y)$ może przyjąc formę
$\Phi(d_m^y) = min_i\{d_m^y(i) : i = 1, \ldots, m\}$

Teoria prawdopodobieństwa

Używana jest teoria prawdpodobieństwa z trzech powodów:

1. pozwala na łatwą generalizację na algorytmy stochastyczne
2. zapewnia proste, spójne ramy, w których można przeprowadzić dowody
3. kluczowym elementem jest rozkład $P(f) = P(f(x_1), ..., f(x_{|X|}))$. Rozkład ten, zdefiniowany na F, daje prawdopodobieństwo, że dany $f \in F$ jest rzeczywistym problemem optymalizacyjnym.

Używając prawdopodobieństwa wydajność algorytmu a po m iteracjach na funkcji kosztu f jest mierzona przy pomocy $P(d_m^y | f, m, a)$.

Twierdzenie pierwsze

Dla każdej pary algorytmów a_1 i a_2

$$\sum_f P(d_m^y | f, m, a_1) = \sum_f P(d_m^y | f, m, a_2)$$

Funkcja kosztu zależna od czasu

Podobny fakt zachodzi dla klasy czasowo zależnych funkcji kosztu. Rozważamy początkową funkcję f_1. Przed rozpoczęciem każdej kolejnej iteracji funkcja kosztu jest przekształcana do nowej funkcji zgodnie z odwzorowaniem $T : F \times \mathbb{N} \to F$. Dane przekształcenie zakładamy, że jest bijekcją i oznaczamy T_i, więc funkcja w i-tej iteracji to $f_{i+1} = T_i(f_i)$.
Możliwe są dwa schematy:
$d_m^y = \{f_1(d_m^x(1)), \ldots, T_{m-1}(f_{m-1})(d_m^x(m))\}$
$D_m^y = \{f_m(d_m^x(1)), \ldots, f_m(d_m^x(m))\}$

Twierdzenie drugie

Dla każdych d_m^y, D_m^y, $m > 1$, algorytmów a_1, a_2 i każdej funkcji początkowej f_1

$$\sum_T P(d_m^y | f_1, T, m, a_1) = \sum_T P(d_m^y | f_1, T, m, a_2)$$

oraz

$$\sum_T P(D_m^y | f_1, T, m, a_1) = \sum_T P(D_m^y | f_1, T, m, a_2)$$

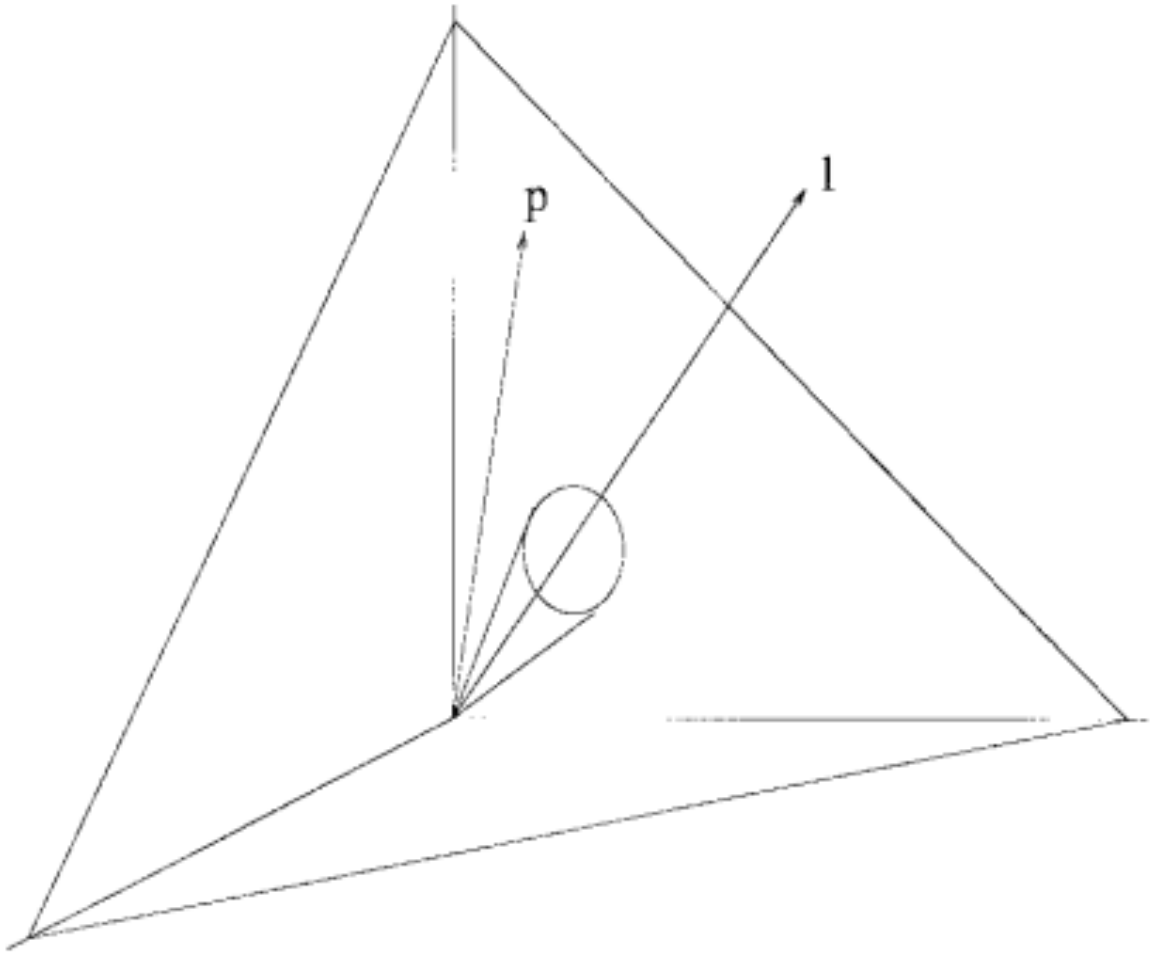
Wniosek

Intuicyjnie, NFL ilustruje, że jeśli wiedza o f, może dana przez $P(f)$, nie jest uwzględniona w a, to nie ma można mieć pewności, że a będzie efektywny. Wówczas efektywna optymalizacja polega na fortunnym zgraniu się f i a.
Wniosek ten jest formalnie uzasadniony przy pomocy geometrycznej reprezentacji.

Reprezentacja geometryczna

Wiemy, że $P(d_m^y|m,a) = \sum_f P(d_m^y|m,a,f)P(f)$. Ową sumę można interpretować jako iloczyn skalarny w F. Oznaczmy wektory $\vec{v}_{d_m^y,a,m}(f) \equiv P(d_m^y|m,a,f)$ i $\vec{p}(f) \equiv P(f)$.
Wówczas $P(d_m^y|m,a) = \vec{v}_{d_m^y,a,m} \cdot \vec{p}$

Reprezentacja geometryczna



Rysunek: [1]

Teoretyczne aspekty optymalizacji

Rozważmy teraz jedynie histogram $\vec{c} = (c_{Y_1}, c_{Y_2}, \ldots, c_{Y_{|Y|}})$, gdzie c_{Y_i} jest liczbą wystąpień Y_i w d_m^y.

Twierdzenie trzecie

Dla dowolnego algorytmu frakcja funkcji kosztu, które skutkują danym histogramem $\vec{c} = m\vec{\alpha}$ to

$$\rho_f(\vec{\alpha}) = \frac{\binom{m}{c_1 c_2 \ldots c_{|Y|}} |Y|^{|X|-m}}{|Y|^{|X|}} = \frac{\binom{m}{c_1 c_2 \ldots c_{|Y|}}}{|Y|^m}$$

Teoretyczne aspekty optymalizacji

Rozważmy histogram $\vec{\beta}$, taki że $N_i = \beta_i|X|$ to liczba punktów w X, dla których $f(x) = Y_i$

Twierdzenie czwarte

Dla danej funkcji f z histogramem $\vec{N} = |X|\vec{\beta}$ frakcja algorytmów dających histogram $\vec{c} = m\vec{\alpha}$ jest równa

$$\rho_{alg}(\vec{\alpha}, \vec{\beta}) = \frac{\prod_{i=1}^{|Y|} \binom{N_i}{c_i}}{\binom{|X|}{m}}$$

Wnioski

1. Nie istnieje uniwersalny algorytm
2. Wstępna wiedza lub założenia o problemie może pozwolić na dobranie odpowiedniego algorytmu
3. Wykonywanie benchmarków jest kluczowe w szukaniu rozwiązania dla danej klasy problemu
4. Zrozumienie problemu jest nawet ważniejsze
5. Wielokrotne optymalizowanie z pozoru podobnych problemów może wymagać wielokrotnego szukania algorytmu

Bibliografia

[1]
David H. Wolpert and William G. Macready (1997)
No free lunch theorems for optimization
IEEE Transactions on Evolutionary Computation 1(1), 67–82.