Articole

CONSIDERAȚII PRIVIND TEOREME DE ECHIVALENȚĂ PENTRU ALGORITMI DE CĂUTARE

ing. Cezar Ionescu
ing. Leon Bațachia

Institutul de Cercetări în Informatică

Rezumat: În acest articol se face o prezentare a teoremelor NFL ("No Free Lunch Theorems") a căror publicare a stârnit multe dezbateri în lumea științifică. Prezentarea este însoțită de considerații privind extinderi, corolare și implicații ale acestor teoreme.
Cuvinte cheie: algoritmi de căutare, optimizare, teoremele NFL.

1. Considerații preliminare

Teoremele de echivalență, prezentate în acest articol, se datorează cercetatorilor americani William Macready și David Wolpert. Articolul în care au fost publicate aceste teoreme este intitulat "No Free Lunch Theorems for Search", denumirea provenind din zicala englezească "there is no such thing as a free lunch". Articolul a fost anunțat în grupul de discuții, rezervat algoritmilor genetici pe Internet, comp.ai.genetic, pe data de 7 februarie 1995, și a generat cea mai lungă listă de discuții din istoria de până acum a grupului respectiv.

Teoremele de echivalență, date de Macready și Wolpert, demonstrează echivalența algoritmilor de căutare în spații discrete (noțiunea de algoritm de căutare va fi formalizată riguros în cele ce urmează), atunci când este considerată media performanței lor pe mulțimea tuturor funcțiilor de cost. Aceasta revine la imposibilitatea existenței unui algoritm mai bun "în medie" decât oricare altul: algoritmii pot fi comparați numai pentru clase de probleme precizate. Este, deci, inutilă încercarea de a găsi un algoritm de căutare "general": toți algoritmii de căutare obțin aceleași performanțe. În particular, algoritmul genetic nu va obține performanțe mai bune decât căutarea aleatoare care, la rândul ei, va avea aceleași performanțe cu metodele de *hill-climbing*.

Macready și Wolpert insistă, în articolul lor, asupra faptului că nici un algoritm de căutare nu poate fi considerat un *panaceu universal* și asupra necesității alegerii unui algoritm în funcție de problema dată (sau a unei reprezentări adecvate pentru problema considerată).

În continuare, articolul este structurat după cum urmează:

- prezentarea notațiilor folosite și formalizarea noțiunii de algoritm de căutare;
- enunțarea și demonstrarea teoremei de echivalență Macready și Wolpert;
- extinderea rezultatelor la algoritmi mai generali;
- principalele corolare ale teoremei;
- discutarea teoremei și a implicațiilor sale.

2. Notații

Vom considera funcții de cost de forma:

$f:X\to Y$, unde X si Y sunt mulțimi finite.

Mulțimea tuturor acestor funcții este Y^X, având cardinalul $|Y|^{|X|}$, unde prin $|A|$ am notat cardinalul mulțimii A.

Un algoritm de căutare ***a*** extrapolează pe baza unei mulțimi de puncte $(x_1,y_1)\ldots(x_m,y_m)$ și propune un nou punct. Extrapolarea poate fi deterministă sau stocastică.

Notăm:

$d^x_m=(x_1,\ldots,x_m)$

$d^y_m=(y_1,\ldots,y_m)$

$d_m=\{d_m(j)\}=\{d^x_m(j), d^y_m(j)\}, j=1\ldots m,$

iar mulțimea tuturor d_m cu D_m. Fie $D=\cup_m D_m$, D conține mulțimea vidă.

Se presupune că punctele (x_j, y_j) sunt distincte, două câte două, și că rezultatul extrapolării algoritmului este, de fiecare dată, un punct nou. Atunci când $m = |X|$, algoritmul se oprește (rezultatul fiind echivalent cu al unei enumerări).

Cu aceste notații, un algoritm de căutare determinist este o funcție

$a{:}d\in D\to\{x|x\notin d^x\}$

Fie c histograma valorilor funcției de cost, obținute de un algoritm ***a*** după m evaluări ale funcției de cost. Avem:

$c \in |N^{|Y|}$, c(i) = numărul de membri din populația d_m ce au valoarea f_i.

Histograma poate fi folosită pentru a aprecia calitatea căutării efectuate de algoritmul a în orice fel. De exemplu, în cazul încercării de a maximiza funcția f, un algoritm este cu atat mai bun cu cât histograma sa contine valori mai numeroase pentru valori f_i mai mari. Ca atare, măsura efectiva a performantei algoritmului depinde exclusiv de histogramă pe care o obține, prin intermediul unei funcții de performanță. Ca atare, ne putem ocupa de histogramele obținute de diverși algoritmi în m pași, fără să ne punem problema criteriului efectiv de performanță al algoritmului (dacă se încearcă minimizarea sau maximizarea funcției).

Valoarea de interes pentru comparatia algoritmilor este, deci:

P(c|f,m,a) (probabilitatea condiționată ca histrogama c să fie obținută după m iterații ale algoritmului a aplicat pentru f).

3. Teorema de echivalență a algoritmilor de căutare (Macready, Wolpert)

Pentru orice algoritmi de căutare a_1, a_2:

$$\sum_f P(c|f,m,a_1) = \sum_f P(c|f,m,a_2)$$

Demonstrație

Vom demonstra că $\Sigma_f P(c|f,m,a)$ nu depinde de a.

Dezvoltând suma după toate componentele y posibile ale unei populații d^y_m avem:

$$\sum_f P(c|f,m,a) = \sum_{f,d^y_m} P(c,d^y_m|f,m,a)$$

Dar

$$P(c,d^y_m|f,m) = P(c|d^y_m,f,m,a)P(d^y_m|f,m,a)$$

iar $P(c|d^y_m, f,m)$ depinde numai de valorile y ale populației d_m. Deci:

$$\sum_f P(c|f,m,a) = \sum_{f,d^y_m} P(c|d^y_m)P(d^y_m|f,m,a) =$$

$$= \sum_{d^y_m} P(c|d^y_m)\sum_f P(d^y_m|f,m,a) \qquad (*)$$

Pentru a demonstra că membrul drept al ecuației (*) este independent de a, este suficient să arătăm că, pentru orice m si d^y_m, $\sum_f P(d^y_m|f,m,a)$ este independentă de a. Vom demonstra aceasta prin inducție.

Pentru m=1, $d_1=\{d^x_1, f(d^x_1)\}$, deci avem:

$$\sum_f P(d^y_1|f,m=1,a) = \sum_f \delta(d^y_1, f(d^x_1))$$

unde δ este funcția Kronecker delta. De aici:

$$\sum_f P(d^y_1|f,m=1,a) = |Y|^{|X|-1}$$

cantitate independentă de a, deci pentru m = 1 propoziția este verificată.

În continuare, demonstrăm ca P(m) ⇒ P(m+1). Avem:

$$P(d^y_{m+1}|f,m+1,a) =$$
$$P(\{d^y_{m+1}(1),\ldots,d^y_{m+1}(m)\},$$
$$d^y_{m+1}(m+1)|f,m+1,a) =$$
$$P(d^y_m, d^y_{m+1}(m+1)|f,m+1,a) =$$
$$= P(d^y_{m+1}(m+1)|d_m,f,m+1,a)$$
$$P(d^y_m|f,m+1,a)$$

și deci:

$$\sum_f P(d^y_{m+1}|f,m+,a) =$$
$$= \sum_f P(d^y_{m+1}(m+1)|d^y_m,f,m+1,a) =$$
$$= P(d^y_m|f,m+,a)$$

Dar $d^y_{m+1}(m+1)$ depinde numai de noua valoare x și de f, deci:

$$\sum_f P(d^y_{m+1}|f,m+,a) =$$
$$= \sum_{f,x} P(d^y_m(m+1)|f,x)P(x|d^y_m,f,m+1,a)$$
$$P(d^y_m|f,m+1,a)$$
$$= \sum_{f,x} \delta(d^y_{m+1}(m+1),f(x))P(x|d^y_m,f,m+1,a)$$
$$P(d^y_m|f,m+1,a)$$

Observăm că $x = a(d^x_m, d^y_m)$ nu depinde în mod direct de f. Rezultă:

$$\sum_f P(d^y_{m+1}|f,m+1,a) = \sum_{f,x,d^x_m} \delta(d^y_{m+1}(m+1),f(x))P(x|d_m,a)$$
$$P(d^x_m|d^y_m,f,m+1,a)P(d^y_m|f,m+a)$$
$$= \sum_{f,d_m} \delta(d^y_{m+1}(m+1),f(a(d_m)))P(d_m|f,m,a)$$

Ținând cont că $a(d^x_m) \notin d^x_m$ avem:

$$\sum_{f} P(d^{y}_{m+1} | f, m+1, a) =$$

$$= \sum_{d^{x}_{m}} \sum_{f(x \in d^{x}_{m})} P(d_{m} | f, m, a) \sum_{f(x \in d^{x}_{m})} \delta(d^{y}_{m+1}(m+1), f(a(d_{m})))$$

Dar $\sum_{f(x \notin d^{x}_{m})} \delta(d^{y}_{m+1}(m+1), f(a(d_{m}))) = |Y|^{|x|-m-1}$

şi deci:

$$\sum_{f} P(d^{y}_{m+1} | f, m+, a) = |Y|^{|X|-m-1} \sum_{f(x \in d^{x}_{m}), d^{x}_{m}} P(d_{m} | f, m, a) =$$

$$= \frac{1}{|Y|} \sum_{f(x \in d^{x}_{m}), d^{x}_{m}} P(d_{m} | f, m, a)$$

$$\frac{1}{|Y|} \sum_{f, d^{x}_{m}} P(d_{m} | f, m, a) = \frac{1}{|Y|} \sum_{f} P(d^{y}_{m} | f, m, a)$$

Din ipoteza de inducţie, membrul drept al acestei relaţii este independent de *a*, deci şi membrul stang este independent de *a*. Cu acestea, teorema este demonstrată.

4. Extinderea teoremei la algoritmi de căutare generali

Teorema a fost demonstrată pentru cazul algoritmilor determinişti care produc de fiecare dată un punct nou din spatiul X. Macready şi Wolpert au arătat însă cum se poate extinde rezultatul obţinut de ei la cazul general al algoritmilor de căutare stocastici şi care pot revizita puncte din X.

Extinderea la cazul algoritmilor ce revizitează puncte din X porneste de la observatia că, din orice 'drum' al unui algoritm *a* ce revizitează puncte, se poate extrage un drum compactat, care pastreaza numai o vizita a fiecarui punct din drumul iniţial. Algoritmii compactaţi sunt supuşi teoremei de echivalenţă şi, ca atare, şi algoritmii originali se supun acestei teoreme.

Trebuie subliniat ca raţionamentul de mai sus este valabil doar în cazul algoritmilor originali care reusesc să exploreze tot spaţiul X, fără să rămână prinşi în "capcane", adică să genereze permanent puncte dintr-o anumită submulţime a lui X.

În mod similar, se pot compacta şi algoritmii stocastici. Atunci, un algoritm stocastic s reprezintă o funcţie definită pe d (ca la algoritmii determinişti) care produce o distribuţie de probabilitate peste X cu proprietatea ca distribuţia este nulă în punctele x $\in$ d. Ca atare, s poate fi privit ca un parametru în funcţia $P(d^{y}_{m+1}(m+1) | d_{m}, s)$ pentru m şi d dăţi.

Cu această definiţie pentru s, dacă urmărim demonstrarea teoremei de echivalenţă, înlocuind peste tot *a* cu s, relaţiile îşi păstrază valabilitatea.

Deci, teorema de echivalenţă este valabilă şi pentru algoritmi stocastici.

5. Corolare ale teoremei de echivalenţă

1. Nu există algoritmi mai rapizi în medie (de exemplu pentru gasirea maximului) decât alţii.

2. Pentru maximizarea funcţiilor, hill-climbing e, în medie, la fel de bun ca hill-descending.

3. Pentru o funcţie de cost oarecare, nu există nici o modalitate de a prevedea performanţa unui algoritm pe baza performanţei anterioare (deci nu se poate prevedea calitatea punctului x_{m+1} pe baza punctelor $x_1, \ldots, x_m$ şi a valorilor funcţiei f aleasă în mod aleator din Y^X în aceste puncte).

Demonstraţie

Orice procedeu de alegere a unei funcţii din Y^X după o distribuţie de probabilitate aleatorie este echivalent cu următorul: pentru orice x $\in$ X se alege un y = f(x) $\in$ Y după o distribuţie uniformă de probabilitate în Y (fiecare y $\in$ Y are o probabilitate de 1/|Y| de a fi ales). Ca atare, valorile funcţiei f sunt independente probabilistic, deci nici un algoritm nu poate prevedea mai bine decât pur aleator valoarea într-un punct nevizitat.

Observaţie

Demonstraţia de mai sus, care ne aparţine, are ca rezultat imediat echivalenţa tuturor algoritmilor de căutare, în medie, cu algoritmul de căutare aleatoare. Deci ea poate fi folosită ca o altă demonstraţie a teoremei Macready şi Wolpert.

6. Critici şi implicaţiile teoremei

Principala critică adusă teoremei NFL a fost ca demonstraţia ei depinde în mod esenţial de o distribuţie de probabilitate uniformă pe mulţimea funcţiilor Y^X. Critică a fost adusă autorilor de către Kihong Park, de la Boston University, dar teorema a fost sprijinită şi de autori mult mai cunoscuţi, cum ar fi dr. Vose, autorul modelării cu Lanţuri Markov a algoritmilor genetici.

Obiecţia ridicată de Kihong Park este următoarea: în aproape toate situaţiile în care lucrăm cu optimizări combinatoriale, avem un mod de a genera funcţia de cost f pe calculator. Or, mulţimea funcţiilor de cost generabile pe calculator este foarte mică în comparaţie cu mulţimea tuturor funcţiilor de la X la Y. Ca atare, nu se poate vorbi despre o distribuţie uniformă pe mulţimea Y^X, pentru ca ştim de la bun început ca probabilitatea asociată celor mai multe funcţii (funcţiilor aproape incompresabile în sens Kolmogorov) este nulă. Aceste funcţii nu pot fi descrise într-un limbaj de programare printr-un program mai scurt decât tabela lor de valori; pentru o funcţie booleană de 300 de variabile s-ar obţine astfel $2^{300} \approx 8*10^{30}$ linii de cod, mai mult decât orice program cunoscut. Or, algoritmii genetici sunt frecvent folosiţi pentru optimizarea funcţiilor booleene de sute de variabile. Deci, conchide Park, nu se poate susţine afirmaţia ca distribuţia de probabilitate este uniformă pe mulţimea funcţiilor şi, ca atare, teorema nu are o relevanţă reală în practică.

Autorii teoremei s-au apărat susţinând ca:

1. Teorema prezintă un fapt matematic real, indiferent de limitările cu care ne confruntăm în viaţă;

   şi

2. Nu există nici un motiv pentru care un rezultat demonstrat pentru multimea tuturor funcţiilor nu ar putea fi aproximativ corect şi pentru o submulţime cum este cea a funcţiilor suficient de puţin complexe în sens Kolmogorov pentru a putea fi implementate pe calculator.

La acestea putem adauga că:

1. Multe aplicaţii ale algoritmilor genetici maximizează o funcţie de cost ce nu este simulată pe calculator (cum ar fi, de exemplu, în aplicaţiile economice, în care valoarea funcţiei pentru o anumită stare este măsurată din indicatori reali);
2. Mulţimea funcţiilor "suficient" de complexe din punct de vedere Kolmogorov nu este bine definită, iar rezultatele caracteristice pentru diverşii algoritmi sunt obţinute pentru mulţimea tuturor funcţiilor definite pe mulţimile X şi Y respective (vezi, de pildă, teorema şablonului).

Cu toate acestea, relevanţa practică a teoremei rămâne discutabilă.

Importanţa teoremei de echivalenţă şi a corolariilor ei constă, însă, în urmatoarele:

1. Au fost expuse formulările eronate de genul "algoritmii genetici reprezintă metode, în general, mai bune decât căutarea aleatoare";
2. S-a stabilit o direcţie clară pentru adaptarea algoritmilor la problema. Până acum, nu a fost evident ce însemna "a cunoaşte o problema", pentru a putea adapta un algoritm la ea. În formularea teoremei NFL, se vede clar ca a cunoaşte o funcţie de cost revine la:

   a) a cunoaşte o distribuţie de probabilitate P(f) după care se extrage funcţia;

   sau, echivalent:

   b) a cunoaşte distribuţiile posibile ale valorilor P(f(x)=y) pentru $y \in Y$.

De altfel, teoremele de convergenţă, demonstrate până acum, relativ la algoritmii genetici cad, în linii mari, în două categorii:

i) teoreme bazate pe o bună adecvare a reprezentării la funcţia de cost, cum este teorema şablonului;

ii) teoreme de convergenţă globală, care se bazează pe proprietatea unui algoritm genetic de a "epuiza" spaţiul X pentru un timp de rulare suficient de mare.

Bibliografie

1. **BAŢACHIA, I. L., C. IONESCU**: Metode de optimizare bazate pe algoritmi genetici. Raport de cercetare, 2064GR/1996.
2. **WOLPERT, D.H., MACREADY, W.G.**: No Free Lunch Theorems for Search. Technical Report, Santa Fe Institute, Argentina, 1995, ftp://ftp. santafe.edu/pub/wgm/nfl.ps.
3. * * *: GA-Digest, Vol. 9-16, http://www.aic.nrl.navy.mil/galist.
4. **MICHALEWICZ, Z.**: Genetic Algorithms + Data Structures = Evolution Programs, Springer Verlag, New York, 1994.