<u>Secvența lui Sturm pentru izolarea rădăcinilor reale</u>

Păduraru Andra – Elena, An3, Grupa A4, Tema 26

Despre

Teorema lui Sturm este denumită după Jacques Charles François Sturm, care a descoperit-o în 1829. În matematică, în special în analiza numerică, izolarea rădăcinilor reale a unui polinom se referă la realizarea unor intervale din mulțimea numerelor reale, unde fiecare conține câte o singură rădăcina reală a polinomului. Mulțimea acestor rădăcini conține toate rădăcinile reale ale polinomului.

Teorema lui Sturm a rezultat prima procedură completă de izolare a rădăcinilor. Aceasta se referă la numărul de rădăcini reale dintr-un interval, având în vedere numărul de variații ale semnului a valorilor din polinom și s-a denumit *Secvența lui Sturm (Sturm's sequence)*. Secvența Sturm al unui polinom uni-variat *p* este o secvență de polinoame asociate lui *p* și derivatei acestuia luându-se în considerare una dintre variantele lui Euclid de rezolvare. Diferența dintre polinoamele uni-variate, multi-variate și bi-variate este dată de numărul de nedeterminați (unul, mai mulți și respectiv doi nedeterminanți).

Secvența lui Sturm este reprezentata de o secvență de rădăcini ce apar în urma aplicării variantei lui Euclid asupra polinomului și derivaților lui. La implementarea algoritmului în computere a fost observat faptul că izolarea rădăcinilor prin teorema lui Sturm nu este la fel de eficientă la fel ca celelalte metode, precum regula semnelor a lui Descartes. Din acest motiv, teorema lui nu este folosită pentru calcule foarte eficiente, decât extrem de rar, dar este foarte utilă din punct de vedere al teoriei.

Izolarea rădăcinilor reale este utilă deoarece, algoritmii de găsire a acestora la un polinom dat, deși vor produce câteva dintre rădăcinile reale, nu vor da cu siguranță toate valorile posibile. De exemplu, dacă nu se găsește nicio rădăcina, nu se poate ști dacă este din cauză că nu exista cu adevărat una sau dacă e din cauză că algoritmul nu a reușit sa le găsească. Unii algoritmi vor găsi toate rădăcinile complexe, dar cum de obicei sunt mult mai puține rădăcini reale decât complexe, majoritatea timpului de execuție este pierdut în găsirea celor complexe. De exemplu, în general un polinom de ordin *n* are *n* rădăcini complexe și *log(n)* rădăcini reale, ceea ce duce la concluzia că nu este optim.

Teorema

Șirul lui Sturm al secvenței acestuia al unui polinom uni-variat *P(x)* este restul împărțirii lui Euclid dintre P_{i-1} și P_i. Fie P(x) polinomul uni-variat cu secvența de polinoame P_0, P_1, ... astfel încât:

- $P_0 = P$
- $P_1 = P'$
- $P_{i+1} = -rem(P_{i-1}, P_i)$

pentru $i \geq 1$, unde P' este prima derivată a lui *P*, iar $rem(P_{i-1}, P_i)$ reprezintă restul de la împărțirea lui P_{i-1} și P_i prin metoda lui Euclid. Lungimea acestei secvențe este cel mult gradul lui *P*.

Numărul variațiilor de semn a secvenței lui Sturm pentru *P* la momentul ξ este egal cu numărul schimbărilor de semn, 0 fiind ignorat, în următoarea secvența de numere reale: $P_0(\xi)$, $P_1(\xi)$, $P_2(\xi)$... Numărul de variații ale semnului este notat cu $V(\xi)$.

Teorema lui Sturm precizează că, dacă *P* este un polinom fără pătrate, numărul de rădăcini reale distincte care se afla în prima jumătate a intervalului (a, b] este egal cu rezultatul lui $V(a) - V(b)$, unde $a, b \in \mathbb{R}$, cu $a < b$. Pentru intervalul $(-\infty, +\infty)$ teorema este extinsă, iar semnul spre $+\infty$ este dat de coeficientul termenului cu cel mai mare grad și cel spre $-\infty$ este dat, în cazul unui polinom de grad par, de același coeficient ca și în cazul precedent, iar dacă polinomul este de grad impar, de opusul semnului coeficientului termenului cu cel mai mare grad. Dacă *P* conține pătrate și dacă nici *a* și nici *b* nu sunt rădăcini multiple ale polinomului, atunci $V(a) - V(b)$ reprezintă numărul de rădăcini distincte reale.

De exemplu: când valoarea lui *x* crește de la *a* la *b* e posibil sa treacă prin valoarea 0 dată de un P_i, cu $i > 0$. În momentele de acest fel, numărul de variații de semn dintre (P_{i-1}, P_i, P_{i+1}) nu se schimbă. Când *x* trece printr-o rădăcina a lui $P_0 = P$, numărul de variații de semn a lui (P_0, P_1) scade de la 1 la 0. Acestea sunt singurele valori pe care *x* le poate lua în așa fel ca semnul să se schimbe.

Exemplu

Se presupune că dorim să găsim numărul rădăcinilor reale dintr-un interval dat pentru următorul polinom: $P(x) = x^4 + x^3 - x - 1$. Vom avea următoarele rezultate:

- $P_0(x) = P(x) = x^4 + x^3 - x - 1$
- $P_1(x) = P'(x) = 4x^3 + 3x^2 - 1$

Restul împărțirii lui Euclid dintre P_0 și P_1 este $-\frac{3}{16}x^2 - \frac{3}{4}x - \frac{15}{16}$. Prin înmulțirea cu -1 se va obține:

- $P_2(x) = \frac{3}{16}x^2 + \frac{3}{4}x + \frac{15}{16}$.

Pentru P_3 și P_4 se va face împărțirea între P_1 și P_2 și înmulțirea restului cu -1, respectiv împărțirea dintre P_2 și P_3 și înmulțirea restului cu -1 și se va obține:

- $P_3(x) = -32x - 64$
- $P_4(x) = -\frac{3}{16}$

Se observa ca $P_4(x) = -\frac{3}{16}$ este o constantă ceea ce înseamnă că algoritmul de căutare a secvenței lui Strum s-a terminat. Pentru găsirea numărului de rădăcini reale ale lui P_0 trebuie evaluată secvența de semne ale polinoamelor la $-\infty$ și $+\infty$, iar aceasta este: $(+, -, +, +, -)$ și $(+, +, +, -, -)$. $V(-\infty) - V(+\infty) = 3 - 1 = 2$, deci P are două rădăcini reale. O altfel de verificare este următoarea: P(x) se poate descompune în $(x^2 - 1) * (x^2 + x + 1)$, unde $(x^2 - 1)$ are rădăcinile -1 și 1, iar $(x^2 + x + 1)$ nu are nicio rădăcină reală, de unde rezultă rezultatul de mai sus.

Generalizarea teoremei

Pentru a defini fiecare polinom din secvența, Sturm a utilizat metoda negativului restului împărțirii lui Euclid dintre ultimele două polinoame. Secvența lui Sturm a fost generalizată în două direcții. O astfel de secvență e o secvență de polinoame cu coeficienți reali P_0, P_1, ... P_m, astfel încât:

1. Întâi gradele scad: $deg(P_i) < deg(P_{i-1})$ pentru i = 2,..m.
2. P_m nu are nicio rădăcina reala sau nu își schimba semnul în apropierea rădăcinilor sale reale
3. Dacă $P_i(\xi) = 0$ pentru $0 < i < m$ și $\xi \in \mathbb{R}$, atunci $P_{i-1}(\xi)*P_{i+1}(\xi) < 0$.

Condiția a treia implică faptul că două polinoame consecutive nu au nicio rădăcină reală în comun. Astfel, secvența lui Strum originală este una generalizată dacă și numai dacă polinomul nu are nicio rădăcina reală multiplă, altfel primele două polinoame din secvență ar avea o rădăcină comună.

La rezolvarea secvenței originale a lui Strum prin împărțirea lui Euclid, e posibil să apară un polinom care să aibă un factor ce nu va fi niciodată negativ, de exemplu: x^2 sau x^2+1. În acest caz, dacă se va continua calculul prin înlocuirea polinomului cu coeficientul acestuia ne-negativ, se va obține o secvență Sturm generalizată. Aceasta poate fi folosită pentru calculul numărului de rădăcini reale, ceea ce poate duce la o simplificare a calculului. Cu toate acestea, în general este dificil de găsit asemenea polinoame, dacă puterea lui *x* nu este pară.

Utilizarea secvenței pseudo-resturilor

În algebra computerelor, polinoamele ce sunt luate în calcul au coeficienți întregi sau pot fi transformați în polinoame cu coeficienți întregi. Secvența lui Sturm pentru un astfel de polinom conține în general polinoame (polinoamele rezultante prin aplicarea regulilor) a căror coeficienți nu sunt întregi. Pentru a evita calculele cu numere ce pot fi exprimate ca fracție a doi întregi, o metoda folosită este înlocuirea metodei lui Euclid prin pseudo-divizare pentru calculul celui mai mare divizor comun al polinomului. Mai exact, se va înlocui secvența rămasă a algoritmului euclidian printr-o secvență a pseudo-resturilor: p_0, .. p_k a polinoamelor, astfel încât există constantele a_i și b_i pentru care b_i*p_{i+1} este de fapt restul împărțirii euclidiene dintre a_i*p_{i-1} și p_i. Diferitele forme ale secvențelor pseudo-resturilor sunt date de alegerea lui a_i și b_i. De exemplu, secvența de resturi a algoritmului euclidian este o secvența de pseudo-resturi cu $a_i = b_i = 1$, oricare ar fi *i*, iar secvența Sturm este o secvența de pseudo-resturi cu $a_i = 1$ și $b_i = -1$, pentru orice *i*.

Izolarea rădăcinilor

Având în vedere cele de mai sus, pentru un polinom izolarea rădăcinilor înseamnă repartizarea acestora în intervale distincte, fiecare având câte un interval. Acest lucru ajută în găsirea rădăcinilor și efectuarea rapidă de calcule (cum ar fi în cazul metodei lui Newton, calcule cu numere complexe).

Pentru izolarea rădăcinilor reale, se pornește de la intervalul (a,b] ce conține toate aceste rădăcini și se calculează V(a) și V(b). Acest interval se împarte în două și se alege *c* la mijlocul lui (a,b]. Calculul lui V(c) va da ca rezultat numărul de rădăcini din (a,c] și (c,b] și se va repeta procesul pentru fiecare sub-interval. Dacă în acest proces se găsește un sub-interval ce nu conține rădăcini, acesta poate fi scos din lista de rezultate finale. Dacă se găsește un sub-interval cu exact o rădăcina, se va opri divizarea lui. Procesul se oprește când vor rămâne doar intervale cu o singura rădăcina în ele.

Aplicări

Secvența generalizata a lui Sturm asigură numărarea rădăcinilor unui polinom când un alt polinom e pozitiv sau negativ, dar fără a le calcula explicit. Dacă se știe un interval al unei rădăcini a primului polinom se asigura automat găsirea semnului celui de-al doilea polinom, fără a se calcula rădăcina.
Fie P(x) și Q(x) două polinoame cu coeficienți reali, astfel încât P și Q nu au nicio rădăcină în comun, iar P nu are rădăcini multiple. Mai exact, P și P'Q sunt polinoame co-prime. Aceasta restricție nu afectează generalitatea modului de calcul, deoarece GCD (functia de calcul al celui mai mare divizor comun) permite reducerea cazului general la unul particular și costul de calcul al secvenței lui Sturm este același cu cel al lui GCD. Fie W(a) numărul de variații a semnului în punctul *a* a unei secvente Sturm generalizate începând cu P și P'Q. Dacă $a, b \in \mathbb{R}$ și $a < b$, atunci $W(a) – W(b)$ este egal cu numărul de rădăcini ale lui P în intervalul (a,b], astfel încât $Q(a) > 0 – (W(a) – W(b))$ și $Q(a) < 0$.

Cod - implementare

Funcția de pornire:

```
int main()
{
	Sturm sturm;
	sturm.set_parameters();
	sturm.show_sturm_sequence(sturm.get_sturm_sequence());
	sturm.show_roots(sturm.get_real_roots());
}
```

Clasa Sturm + funcții:

```
constexpr int max_order = 12;

class Sturm
{
	// reprezentarea unui polinom
	typedef struct Polynomial
	{
	   public:
		int ord;
		double coef[max_order];
	} Poly;

	Poly sturm_seq[max_order]; //definire secvența de polinom cu ordin maxim = max_order
	int order, num_poly, nroots, nchanges, atmin, atmax;
	double min = -1.0, max = 1.0, roots[max_order];

	int build_sturm(); //creare secventa
	int modp(); //calcul rest
	int num_roots(); obtinere numar de radacini distincte
	int num_changes(); //obtinere numar de schimbari de semn
	double eval_poly(int, double *, double); //obtinere valoare polinom

   public:
	void set_parameters(); //setare parametri, coefiscienti și ordin
	std::vector<std::vector<double>> get_sturm_sequence(); //obtinere secvența strum
	void show_sturm_sequence(const std::vector<std::vector<double>> &); //vizualizare secventa
	std::vector<double> get_real_roots(); //obtinere radacini reale
	void show_roots(const std::vector<double> &); //vizualizare radacini
};
```

```
void Sturm::set_parameters()
{
	//ordinul polinomului
	cin >> order;

	for (auto i = order; i >= 0; --i)
		//citire coeficienți pentru primul polinom de start
		cin >> sturm_seq[0].coef[i];
}


//afișare secventa
void Sturm::show_sturm_sequence(const std::vector<std::vector<double>> &seq)
{
	auto first = true;

	//pentru fiecare polinom din secvența se afișează coeficienții
	for (const auto &poly : seq)
	{
		for(const auto &coef : poly)
			cout << coef << ' ';
		if (first)
			first = false;
	}
}


//afișare rădăcini
void Sturm::show_roots(const std::vector<double> &roots)
{
	if (roots.size() == 1)
		cout << "\n1 distinct real root at x ";
	else {
		cout << "\n" << roots.size() << " distinct real roots for x: ";
		for (const auto& root : roots)
			cout << root << ' ';
	}
}


//creare secventa
std::vector<std::vector<double>> Sturm::get_sturm_sequence()
{
	std::vector<std::vector<double>> seq;
	std::vector<double> coefs;

	//creare secvența sturm pentru polinom și inițializarea cu numărul de polinoame din secvență
	num_poly = build_sturm();

	//creare vector coeficienți
	for (auto i = order; i >= 0; --i)
		coefs.push_back(sturm_seq[0].coef[i]);

	//creare secvență
	seq.push_back(coefs);
	for (auto i = 0; i <= num_poly; ++i)
	{
		coefs.clear();
		for (auto j = sturm_seq[i].ord; j >= 0; --j)
			coefs.push_back(sturm_seq[i].coef[j]);
		seq.push_back(coefs);
	}
	return seq;
}
```

```
//obținere număr rădăcini reale
std::vector<double> Sturm::get_real_roots()
{
	nroots = num_roots();
	if (nroots == 0)
		cout << "solve: no real roots\n";


	// calcul interval a, b pentru rădăcini
	// împărțirea intervalului a, b
	//calcul vector radacini
}

creare secvența sturm pentru polinom și returnare numărul de polinoame din secventa
int Sturm::build_sturm()
{
	//calcul derivata și normalizare coeficienți
	//calcul rest euclidian
	//calcul semn
}
//obtinere numar de radaciini reale distincte
int Sturm::num_roots() {}

//obtinere numărul de schimbari a semnului
int Sturm::num_changes() {}

//calcul polinom
double Sturm::eval_poly(int ord, double *coef, double x)
{
	double *fp, f;
	fp = &coef[ord];
	f = *fp;
	for (fp--; fp >= coef; fp--)
		f = x * f + *fp;
	return f;
}
```

Bibliografie

http://www.kurims.kyoto-u.ac.jp/~kyodo/kokyuroku/contents/pdf/0986-22.pdf
https://en.wikipedia.org/wiki/Sturm%27s_theorem
https://en.wikipedia.org/wiki/Polynomial#univariate
https://en.wikipedia.org/wiki/Polynomial_greatest_common_divisor#Euclid's_algorithm
https://en.wikipedia.org/wiki/Real-root_isolation
https://en.wikipedia.org/wiki/Square-free_polynomial
https://en.wikipedia.org/wiki/Polynomial_greatest_common_divisor#Pseudo-remainder_sequences