

Gradiento metodai

Optimizavimo uždavinyje ieškomas tikslo funkcijos $f(x)$ maksimumas arba minimumas, pasinaudojant iteracinio proceso algoritmu

$$x^{k+1} = x^k + \alpha_k p^k, \quad k = 1, 2, \dots$$

kur p^k yra žingsnio krypties vektorius, o α_k - žingsnio ilgis.

Jeigu funkcija diferencijuojama, tai pasinaudojus dalinių funkcijos išvestinių reikšmėmis, galima prognozuoti kryptį į ieškomąjį maksimumą arba minimumą.

Būtent funkcijos $f(x)$ gradientas $\nabla f(x^0)$ apibūdina funkcijos greičiausio augimo kryptį ir dydį taške x^0 .

Analogiškai, kryptis – priešinga gradiento kryptčiai – antigradientas $-\nabla f(x^0)$, yra funkcijos $f(x)$ greičiausio mažėjimo (nuolydžio) kryptis taške x^0 .

Todėl funkcijos optimizavimo metodai, kuriuose krypties vektorių nusako gradientas, vadinami gradiento (arba greičiausio nuolydžio, arba greičiausio nusileidimo) metodais.

Gradiente metodo algoritmas

Metodo iteracinè formulè minimumo radimui

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k), \quad k = 1, 2, \dots$$

arba koordinatine forma:

$$x_1^{k+1} = x_1^k - \alpha_k \frac{\partial f(x^k)}{\partial x_1};$$

$$x_2^{k+1} = x_2^k - \alpha_k \frac{\partial f(x^k)}{\partial x_2};$$

...

$$x_n^{k+1} = x_n^k - \alpha_k \frac{\partial f(x^k)}{\partial x_n};$$

$$k = 1, 2, \dots$$

Žingsnio ilgis α_k parenkamas dviem būdais.

1 būdas.

1. Parenkamas skaičius $\alpha > 0$ visiems k ir imamas

$$x = x^k - \alpha \nabla f(x^k);$$

2. Apskaičiuojama

$$f(x) = f(x^k - \alpha \nabla f(x^k));$$

3. Tikrinama nelygybė

$$f(x) - f(x^k) \leq \varepsilon \alpha \|\nabla f(x^k)\|^2$$

čia ε - laisvai pasirinkta konstanta, ta pati visiems k ir $0 < \varepsilon < 1$;

4. Jeigu nelygybė teisinga, tai imamas skaičius

$$\alpha_k = \alpha$$

Jeigu nelygybė neteisinga, tai α keičiamas mažesniu (α dauginamas iš skaičiaus λ , $0 < \lambda < 1$) ir viskas kartojama, kol bus teisinga 3 nelygybė.

2 būdas.

Žingsnio ilgis α_k randamas pagal formulę

$$f\left(x^k - \alpha_k \nabla f\left(x^k\right)\right) = \min_{\alpha \geq 0} f\left(x^k - \alpha \nabla f\left(x^k\right)\right).$$

Gradiento metodas sąlyginiam optimizavimo uždaviniui

Apibrėžimas. Turime erdvėje E^n netuščią uždara aibę S . Erdvės E^n taško A atstumu $\rho(A, S)$ nuo jo iki netuščios uždarnosios aibės S vadinamas mažiausias iš atstumų tarp A ir tos aibės taškų

$$\rho(A, S) = \min_{x \in S} \rho(A, x).$$

Apibrėžimas. Aibės S taškas, kurio atstumas iki taško A lygus atstumui nuo A iki S , vadinamas taško A *projekcija* aibėje S ir žymimas $P_S(A)$.

Algoritmas

Turime sąlyginį minimizavimo uždavinį su diferencijuojama funkcija $f(x)$ ir uždara iškiląja leistinųjų reikšmių aibe X . Jeigu minimumo taškas x^* yra aibės X viduje, tai galima jo ieškoti gradiento metodu:

1. Randamas pradinis aibės X taškas x^0 ;
2. Atliekamos iteracijos

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k), \quad k = 1, 2, \dots;$$

3. Žingsnio ilgis α_k randamas vienu iš aukščiau aprašytų būdų, tačiau turi būti patenkinta sąlyga:

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k) \in X,$$

4. Jei pasirinktam α gauname, kad $x^{k+1} \notin X$, tai α mažiname daugindami iš skaičiaus λ , $0 < \lambda < 1$.

5. Jei minimumo taškas x^* yra kraštinis aibės X taškas arba nežinoma, kur jis yra, tai radus tašką $x^0 \in X$, toliau atliekamos iteracijos

$$x^{k+1} = P_X(x^k - \alpha_k \nabla f(x^k)).$$