Нелинейна оптимизация

# „Множител на Лагранж“

Описание на метода и стъпките, с които се реализира:

* Този метод ни позволява да намираме максимум или минимум на функция на много променливи **f(x1,…,xn)** при някакво ограничение на входните стойности.
* Ограничението е от тип равенство **g(x1,…, xn) = c**, където **“g”** е друга функция със същите аргументи като **“f”**, а **“c”** е константа.
* Основната идея е да намерим точките, където контурните линии на **“f”** и **”g”** се допират една до друга.  
  В тези точки векторите на са перпендикулярни на двете контурни линии. Насочени са в едно направление и реално съвпадат като са пропорционални един на друг.  
  Следователно тяхното приравняване налага и въвеждането на нова константа **λ**.  
  **λ** – Множител на Лагранж

Градиентите са частни производни на функциите по всеки от аргументите им, от където ще получим система с **“n”** уравнения. Към нея ще добавим самото ограничение, за да допълним до система с равен брой уравнения и неизвестни, тоест **“n + 1”**.

След като решим системата ще получим различни стойности за търсените променливи на функцията **f(x1,…,xn)**.  
Замествайки с различните комбинации от променливи във функцията ще получим различни решения, което от тях даде най-висока (най-ниска) стойност, ще бъде търсеният от нас максимум (минимум).

* Всички уравнения от системата в предната точка могат да бъдат капсулирани в едно единствено, което изглежда така:  
  Функцията **L** се нарича „Лагранжиан“ и има следният вид:
* Забележка: В някои източници може да срещнете **λ** с противоположен знак:

Това не води до никаква разлика по отношение решаването на проблема, но го имайте в предвид.

* Възможно е да са зададени и повече от едно ограничения, **“m”** на брой. В този случай се въвеждат и толкова на брой константи **λ**.  
  Тогава функцията **L** ще придобие следният вид:

## „Условия на Каруш – Кун – Такър (ККТ) за оптималност“

* Тези условия позволят използването на ограничения от тип неравенство:

Те обобщават метода на Лагранж, който е само за ограничения от тип равенство.

* Проблем за нелинейна оптимизация:  
  Намерете ***min / max f(x)*** при ограничения , където

целевата функция ***f: Rn –> R*** и ограниченията

***gi: Rn –> R*** садефинирани и непрекъснато диференцируеми функции.

* **Необходими условия**  
  Нека точка ***x\**** е локален оптимум на поставения проблем. Тогава съществуват константи

, наричани ККТ множители, такива че следните условия да бъдат изпълнени:

1. **За *min f(x)*:**

**За *max f(x)*:**

* **Достатъчни условия**Нека удовлетворяват условията (1) – (4). Нека ***f*** и са диференцируеми изпъкнали функции. Тогава точката ***x\****е глобален оптимум на поставения проблем.

**Използвани източници:**

* Wikipedia: [Lagrange multiplier](https://en.wikipedia.org/wiki/Lagrange_multiplier) & [KKT conditions](https://en.wikipedia.org/wiki/Karush%E2%80%93Kuhn%E2%80%93Tucker_conditions)
* Khan Academy: [Lagrange multiplier](https://www.khanacademy.org/math/multivariable-calculus/applications-of-multivariable-derivatives/lagrange-multipliers-and-constrained-optimization/v/constrained-optimization-introduction)
* YouTube: [KKT optimality conditions](https://www.youtube.com/watch?v=UD0ac9pKopk)

### Реализация на алгоритъма чрез

### “Gekko library”

1. Инсталиране на библиотеката:

**!pip install gekko**

1. Импортване:

**from gekko import GEKKO**

1. Създаване и инициализация на обект, посредством който използваме библиотеката:

**n = GEKKO()**

1. Създаване на променливи (Търсените оптимизирани параметри на заема):

* **x1 = n.Var(loanAmount0, 200.0, 160000.0)**

**loanAmount0** – Начална стойност т.е. в нашият случай желаната сума от клиента

1. – Долна граница

**160000.0** – Горна граница

* **x2 = n.Var(loanPeriod0, 2.0, 60.0)**

**loanPeriod0** – Начална стойност т.е. в нашият случай желаният период на изплащане (месеци) от клиента

**2.0** – Долна граница

1. – Горна граница
2. **n.Equation(predictedGood0 - par[0] \* loanAmount0 - par[1] \* loanPeriod0 + par[0] \* x1 + par[1] \* x2 >= cutOff)**

В скора се заместват получените нови

оптимизирани параметри на заема на мястото на

старите - исканите от клиента. В случай, че

неравенството е изпълнено т.е. новият скор не е

по-малък от граничния, на клиента се предлагат

новите параметри на заема. Ако не е изпълнено

библиотеката хвърля грешка т.е. няма решение.

1. Целева функция:

**n.Obj(ϕ)**

1. Решаване на задачата:

**n.solve(disp=False)**

1. За достъпване на оптимизираните стойности се използват:

**x1.value[0]**

**x2.value[0]**