

ML OPT

GeoComput & ML

2021-05-06 Thur

*ML often cast as an optimisation problem,
the problem of determining an argument for which
a given function has extreme values on a given domain*

Optimisation

Given a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and a set $S \subseteq \mathbb{R}^n$, we seek \mathbf{x}^* such that f attains minimum at \mathbf{x}^* , $\forall \mathbf{x} \in S$

We call f the objective function, S the feasible set and any vector $\mathbf{x} \in S$ a feasible point.

Generally, a continuous optimisation problem takes the form

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{g}(\mathbf{x}) = \mathbf{o} \text{ and } \mathbf{h}(\mathbf{x}) \leq \mathbf{o}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, and $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^p$.

Existence and Uniqueness

If f is continuous on a close and bounded set $S \subseteq R^n$, then f has a global minimum on S

Coercive functions

A continuous function f on an unbounded set $S \subseteq R^n$ is said to coercive if

$$\lim_{\|x\| \rightarrow \infty} f(x) = +\infty$$

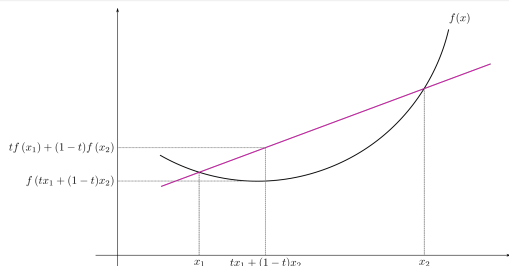
If f is coercive on a close, unbounded set $S \subseteq R^n$, then f has a global minimum on S

Definition

If a function f satisfies the condition

$$f(\alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2) \leq \alpha f(\mathbf{x}_1) + (1 - \alpha) f(\mathbf{x}_2)$$

$$\forall \alpha \in [0, 1] \cup \forall \mathbf{x}_i \in S$$



Any local minimum of a convex function f on convex set $S \subseteq \mathbb{R}^n$ is a global minimum of f on S

1st order necessary condition

$$\nabla f(\mathbf{x}^*) = 0$$

where gradient $\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \dots \quad \frac{\partial f}{\partial x_n} \right)^T$ and \mathbf{x}^* is called a **critical point** of f .

2nd order sufficient condition

At a critical point \mathbf{x}^* , if $\mathbf{H}_f(\mathbf{x}^*)$ is

- positive definite, then \mathbf{x}^* is a minimum.
- negative definite, then \mathbf{x}^* is a maximum.
- indefinite, then \mathbf{x}^* is a saddle point.

where

$$\mathbf{H}_f(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 \mathbf{x}}{\partial x_1^2} & \frac{\partial^2 \mathbf{x}}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 \mathbf{x}}{\partial x_1 \partial x_n} \\ \frac{\partial^2 \mathbf{x}}{\partial x_2 \partial x_1} & \frac{\partial^2 \mathbf{x}}{\partial x_2^2} & \cdots & \frac{\partial^2 \mathbf{x}}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \mathbf{x}}{\partial x_n \partial x_1} & \frac{\partial^2 \mathbf{x}}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 \mathbf{x}}{\partial x_n^2} \end{pmatrix}$$

Lagrange multiplier

Consider problem

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{g}(\mathbf{x}) = \mathbf{o}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, and $m \leq n$.

Define the *Lagrangian function*, $\mathcal{L} : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$.

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{g}(\mathbf{x})$$

then the necessary condition for a critical point is

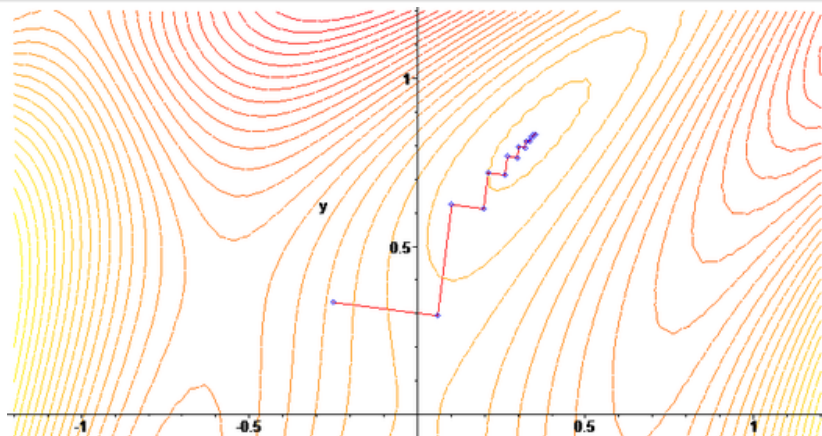
$$\nabla \mathcal{L} = \begin{pmatrix} \nabla f(\mathbf{x}) + \mathbf{J}_{\mathbf{g}}^T(\mathbf{x}) \boldsymbol{\lambda} \\ \mathbf{g}(\mathbf{x}) \end{pmatrix} = \mathbf{o}$$

Unconstrained Optimisation

Gradient Descent

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{s}_k$$

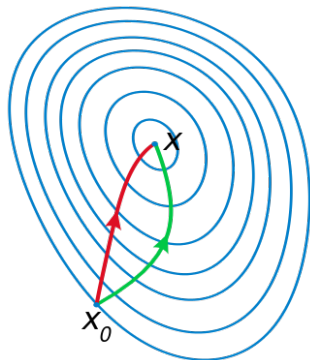
$$\mathbf{s}_k = -\nabla f(\mathbf{x}_k)$$



Newton Method

$$f(\mathbf{x} + \mathbf{s}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{s} + \mathbf{s}^T \mathbf{H}_f(\mathbf{x}) \mathbf{s} / 2$$

$$\mathbf{H}_f(\mathbf{x}) \mathbf{s} = -\nabla f(\mathbf{x})$$



learn a decision function $h : \mathbf{U} \rightarrow \mathbf{V}$

$$\begin{cases} h(\mathbf{u}) = \sum_{i=0}^n \theta_i u_i = \boldsymbol{\theta}^T \mathbf{u} \\ u_0 = 1 \end{cases}$$

search for $\boldsymbol{\theta} \in \mathbb{R}^{n+1}$

$$\min_{\boldsymbol{\theta}} \{f(\boldsymbol{\theta}) := \sum_i (h_{\boldsymbol{\theta}}(u_i) - v_i)^2\}$$

$$\begin{aligned} f(\boldsymbol{\theta}) &= (\mathbf{U}\boldsymbol{\theta} - \mathbf{v})^T (\mathbf{U}\boldsymbol{\theta} - \mathbf{v}) \\ &= \boldsymbol{\theta}^T \mathbf{U}^T \mathbf{U} \boldsymbol{\theta} - 2\boldsymbol{\theta}^T \mathbf{U}^T \mathbf{v} + \mathbf{v}^T \mathbf{v} \end{aligned}$$

$$f'(\boldsymbol{\theta}) = \mathbf{U}^T \mathbf{U} \boldsymbol{\theta}^T - \mathbf{U}^T \mathbf{v} = 0$$

$$\boldsymbol{\theta}^* = (\mathbf{U}^T \mathbf{U})^{-1} \mathbf{U}^T \mathbf{v}$$

Let $\epsilon_i = v_i - \boldsymbol{\theta}^T u_i$ and $\epsilon_i \sim \mathcal{N}(\mu, \sigma^2)$

$$p(v_i | u_i; \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(v_i - \boldsymbol{\theta}^T u_i)^2}{2\sigma^2}\right)$$

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}) &= \sum_i \log\left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(v_i - \boldsymbol{\theta}^T u_i)^2}{2\sigma^2}\right)\right) \\ &= N \log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) - \frac{1}{2\sigma^2} \sum_i (v_i - \boldsymbol{\theta}^T u_i)^2\end{aligned}$$

Logistic Model

Given $\mathbf{U} = (u_1, \dots, u_n)^T \cup \mathbf{v} \in \{0, 1\}$

decision function $h_{\theta} = g(\theta^T u) = \frac{1}{1 + \exp(-\theta^T u)}$

$v_i \sim \mathcal{B}(p)$

$$\mathcal{L}(\theta) = \sum_i \log(h(u_i)^{v_i} (1 - h(u_i))^{1-v_i})$$









$$\max_{\theta} \sum_i (-\log(1 + \exp(-\theta^T u)) - (1 - v_i)\theta^T u)$$

$$\begin{aligned}\hat{\theta} &= \max_{\theta} p(\theta|\mathcal{X}) \\ &= \max_{\theta} p(\mathcal{X}|\theta)p(\theta)\end{aligned}$$

- ML and optimisation
- Intelligence

- time frame
- my expectations

References

-  M. Heath. Scientific Computing An Introductory Survey (2018)
-  C. Aggarwal. Linear Algebra and Optimization for Machine Learning (2020)
-  S. Theodoridis. Machine Learning : A Bayesian and Optimization Perspective (2020)
-  S. Sar et. al. Optimization for Machine Learning (2012)
-  J. Gallier. Fundamentals of Optimization Theory With Applications to Machine Learning (2019)
-  https://en.wikipedia.org/wiki/Convex_function
-  https://en.wikipedia.org/wiki/Gradient_descent
-  https://en.wikipedia.org/wiki/Newton%27s_method_in_optimization
-  https://en.wikipedia.org/wiki/Bernoulli_distribution