Lecture 15 Continuous Optimization I: Gradient Descent	L15.1 6.046 04/11/2017

Today

- Continuous Optimization
- Gradient Descent Method
- Convexivity

Many of the problems we consider this semester (and in 6.006) involve algorithms operating on a discrete (and finite) set of objects.

Interval scheduling ← discrete set of events
Median finding ← discrete set of elements
Graph algorithms ← discrete sets of vertices and edges (although flows could, in principle, be continuous)

Here we consider optimization in continuous spaces, as we did with linear programming (LPs), but with greater generality (can be non-linear).

Examples:

① Find the optimum date and time to launch a space vehicle from Kennedy Space Center so that the time (or fuel, or some function of time and fuel) to reach Mars is minimized.

② Find the optimum combination of funds a given company should spend on celebrity endorsement, advertising, product improvement, and manufacturing enhancement to maximize profit over the next two years.

③ Find the optimum set of chemical properties a molecule should have to bind tightly to a particular disease target, as one step of the lead development process for a drug development campaign.

minimize binding free energy

④ Train a deep learning network model to identify patients at risk for sudden cardiac arrest. Take data from their electronic medical record, and minimize error on training data.

Note: Each involves a set of "control variables" (date and time, funds in multiple categories, multiple chemical properties, weights and biases) that are continuous and multi-dimensional (usually).

- Each has a continuous, scalar objective function that is generally a non-linear function of the control variables.
- A model of some type is usually required to specify the relationship between the control variables and the objective function.

These classes of problems occur frequently in scientific computing, financial modeling, robotics navigation, control, etc.

Problem Statement: Unconstrained Minimization

Given a real-valued function $f: \mathbb{R}^n \to \mathbb{R}$, find its minimum, assuming it exists.

That is, find $\vec{x}^* = \arg\min_{x \in \mathbb{R}^n} f(\vec{x})$

and $f(\vec{x}^*) = \min_{\vec{x} \in \mathbb{R}^n} f(\vec{x})$

objective function

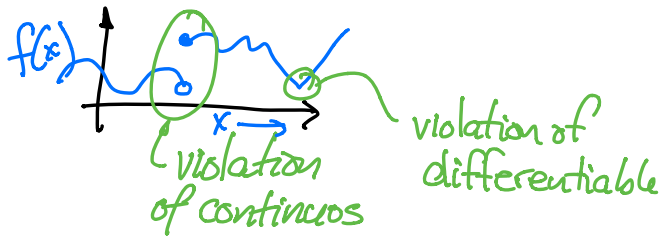
The minimum objective value $f(\vec{x}^*)$ occurs at the point $\vec{x}^*$

Note:

- If we require the maximum of $f(\vec{x})$, just take the minimum of $g(\vec{x}) = -f(\vec{x})$
- If we require constraints, minimize $g(\vec{x}) = f(\vec{x}) + \psi(\vec{x})$ with $\psi(\vec{x}) = 0$ except when constraints violated, where $\psi(\vec{x}) \to \infty$.

Assumptions:

f is continuous and infinitely differentiable



Gradient Descent Method

Iterative approach of locally using gradient to continuously attempt to make improvements by walking downhill.

This is a locally greedy approach.

Can guarantee $f(\vec{x}^{(i+1)}) \leq f(\vec{x}^{(i)})$.

Perform linear expansion about the current point $\vec{x}$:

$$f(\vec{x}+\vec{\delta}) = f(\vec{x}) + \left[\vec{\nabla} f(\vec{x})\right]^T \vec{\delta} + \frac{1}{2}\vec{\delta}^T \vec{\nabla}^2 f(\vec{x})\vec{\delta} + \dots$$

linear approximation; error of that approx.

$\vec{x}, \vec{\delta} \in \mathbb{R}^n$

$$\vec{\nabla} f(\vec{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\vec{x}) \\ \frac{\partial f}{\partial x_2}(\vec{x}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\vec{x}) \end{bmatrix}$$

gradient of f evaluated at $\vec{x}$.

[The direction of greatest local increase in f; so we move in the opposite direction, $-\vec{\nabla} f(\vec{x})$]

inner product or "dot product"

$$\vec{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \Rightarrow \vec{v}^T = \begin{bmatrix} v_1, v_2, \dots, v_n \end{bmatrix}$$

$$\vec{u}^T \vec{v} = \sum_{i=1}^{n} u_i v_i = \langle \vec{u}, \vec{v} \rangle$$

$$\nabla^2 f(\vec{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(\vec{x}) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(\vec{x}) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(\vec{x}) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(\vec{x}) & \frac{\partial^2 f}{\partial x_2 \partial x_2}(\vec{x}) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(\vec{x}) \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(\vec{x}) & \frac{\partial^2 f}{\partial x_n \partial x_2}(\vec{x}) & & \frac{\partial^2 f}{\partial x_n \partial x_n}(\vec{x}) \end{bmatrix}$$

Hessian

Gradient Descent algorithm

Begin at some starting point $\vec{x}^{(0)}$

For $i = 0, 1, \ldots$ $\quad \vec{x}^{(i+1)} = \vec{x}^{(i)} - \eta_i \vec{\nabla} f(\vec{x}^{(i)})$

The condition for local optimality is $\vec{\nabla} f(\vec{x}) = 0$, which could be local minimum, local maximum, or saddle point. This algorithm doesn't provide any improvement from a local optimum, but can locally perturb & continue from max. or saddle.

Choice of step size, η_i

$$f(\vec{x}+\vec{\delta}) \approx f(\vec{x}) + \left[\vec{\nabla} f(\vec{x})\right]^T \vec{\delta} + \frac{1}{2}\vec{\delta}^T \nabla^2 f(\vec{x}) \vec{\delta} + \cdots$$

quadratic term goes as $\|\vec{\delta}\|^2$

We have truncated the expansion at the quadratic term. This will be a good approximation for cases in which the higher-order terms are small with respect to the others, and small step sizes lead to more rapid falloff of higher-order terms.

Key observation: No matter how complex our function (and how rapidly $f(\vec{x})$ changes for a given change in $\vec{x}$), if one examines a small enough local neighborhood of $\vec{x}$, the function behaves very simply — it behaves as if it is linear — and so we can find downhill & step in that direction.

f is β-smooth $\iff$ $\delta^T \nabla^2 f(\vec{x}) \delta \leq \beta \|\delta\|^2 \quad \forall \vec{x}, \delta \in \mathbb{R}^n$

Plugging in our choice of step in gradient descent, $-\eta \vec{\nabla} f(\vec{x}) = \delta$

$$f(\vec{x}+\delta) \leq f(\vec{x}) - \eta \left\| \vec{\nabla} f(\vec{x}) \right\|^2 + \frac{\beta \eta^2}{2} \left\| \vec{\nabla} f(\vec{x}) \right\|^2$$

progress expected — error expected

For β-smooth function f, $f(\vec{x}+\delta) \leq f(\vec{x}) + \delta^T \vec{\nabla} f(\vec{x}) + \frac{\beta}{2}\|\delta\|^2$

If we set our expected progress greater than or equal to the error we are willing to accept:

$$\eta \left\| \vec{\nabla} f(\vec{x}) \right\|^2 \geq \frac{\beta \eta^2}{2} \left\| \vec{\nabla} f(\vec{x}) \right\|^2$$

$$\eta \leq \frac{2}{\beta} \longrightarrow \text{if we choose } \eta \approx \frac{1}{\beta},$$

progress should exceed error in estimated progress.

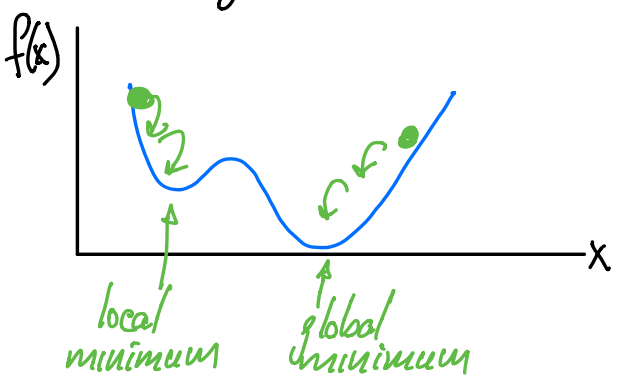
$$\Rightarrow f\left(\vec{x}^{(i+1)}\right) \leq f\left(\vec{x}^{(i)}\right) - \frac{1}{2\beta} \left\| \vec{\nabla} f\left(\vec{x}^{(i)}\right) \right\|^2$$

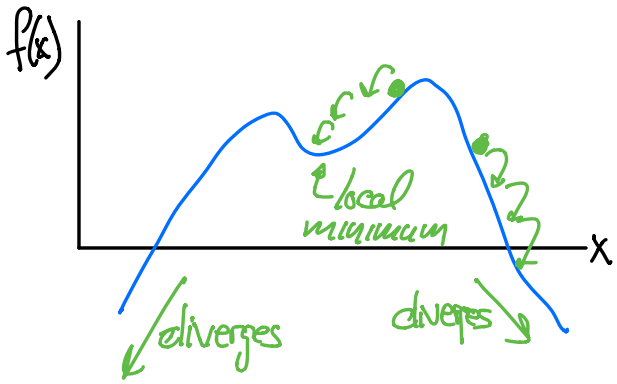
expected minimum progress in i^{th} step

Gradient descent either converges to a point $\vec{x}$ where $\vec{\nabla} f(\vec{x}) = 0$ or it diverges to $f(\vec{x}) \to -\infty$

← local minimum, maximum, or saddle point

Even if it is a local minimum, it may not be a global minimum.





The minimum you end up in depends on where you start.

Gradient descent is guaranteed to converge to a global minimum when the function $f(\vec{x})$ is convex.

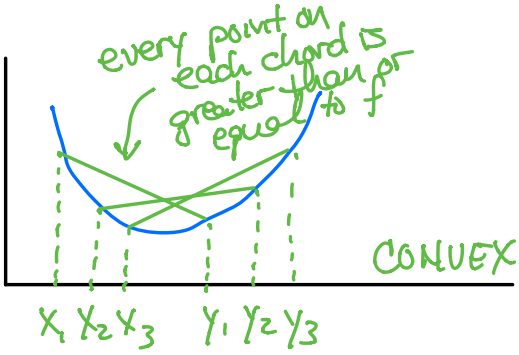
Def: The function $f(\vec{x})$ is convex iff

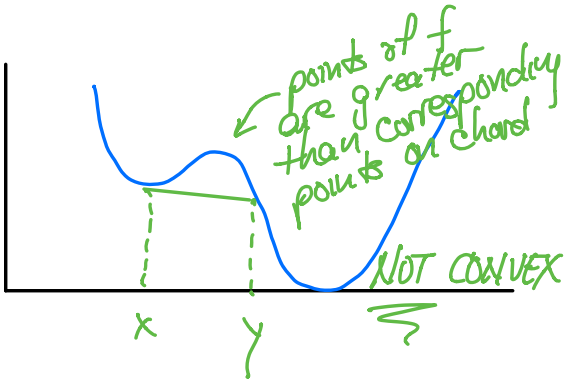
$$\forall_{0 \le \lambda \le 1,\ \vec{x}, \vec{y} \in \mathbb{R}^n} \quad f(\lambda \vec{x} + (1-\lambda)\vec{y}) \le \lambda f(\vec{x}) + (1-\lambda) f(\vec{y})$$

or, equivalently,

$$\forall_{\vec{x}, \vec{y} \in \mathbb{R}^n} \quad f(\vec{x} + \vec{\delta}) \ge f(\vec{x}) + [\vec{\nabla} f(\vec{x})]^T \vec{\delta}$$

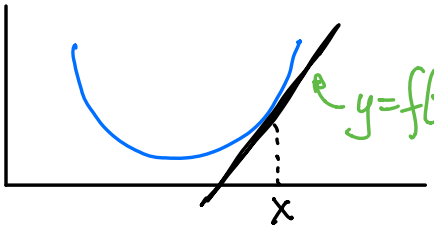
$\vec{\nabla} f(\vec{x}^*) = 0 \Leftarrow \Rightarrow \vec{x}^* = \arg\min_{\vec{x}} f(\vec{x})$





$$f(\vec{x}+\vec{\delta}) \geq f(\vec{x}) + [\vec{\nabla} f(\vec{x})]^T \vec{\delta}$$

← The gradient defines a hyperplane tangent to $f(\vec{z})$ at $\vec{x}$ and that lowerbounds $f(\vec{z})$.



Unfortunately, many important problems, including deep learning, correspond to hypersurfaces that are not convex.

Convergence Analysis — How fast does gradient descent converge?

Let $\vec{x}^*$ be the minimum of convex function f.

$$f(\vec{x}^*) \geq f(\vec{x}) + \left[\vec{\nabla} f(\vec{x})\right]^T (\vec{x}^* - \vec{x})$$

$$\Rightarrow f(\vec{x}) - f(\vec{x}^*) \leq -\left[\vec{\nabla} f(\vec{x})\right]^T (\vec{x}^* - \vec{x}) \leq \left\|\vec{\nabla} f(\vec{x})\right\| \cdot \left\|\vec{x}^* - \vec{x}\right\|$$

How far we are from minimum function value

Cauchy-Schwartz Inequality

$$\Rightarrow \text{If } \left\|\vec{\nabla} f(\vec{x})\right\| \leq \frac{\varepsilon}{\left\|\vec{x}^* - \vec{x}\right\|}, \text{ then the function value } f(\vec{x}) \text{ is at most } \varepsilon \text{ away from the optimum } f(\vec{x}^*).$$

This bound on how close we are to optimal depending on the unknown distance between $\vec{x}$ and $\vec{x}^*$ is unfortunate but inherent — we really don't know where the optimum is.

We can improve our situation for a particular class of convex functions, those that possess α-strong convexity.

Def: $f(\vec{x})$ is α-strongly convex iff

$$\vec{y}^T \nabla^2 f(\vec{x})\, \vec{y} \geq \alpha \left\|\vec{y}\right\|^2 \quad \forall \vec{x}, \vec{y} \in \mathbb{R}^n \text{ for } \alpha > 0$$

(Normal convexity corresponds to $\alpha = 0$)

For functions that possess α-strong convexity,

$$f(\vec{x}+\vec{\delta}) \geq f(\vec{x}) + [\vec{\nabla} f(\vec{x})]^T \vec{\delta} + \frac{\alpha}{2}\|\vec{\delta}\|^2$$

Quadratic Function that lower bounds f

Considering our convergence to minimum:

$$f(\vec{x}) \geq f(\vec{x}^*) + \underbrace{[\vec{\nabla} f(\vec{x}^*)]^T}_{\equiv 0}(\vec{x}-\vec{x}^*) + \frac{\alpha}{2}\|\vec{x}-\vec{x}^*\|^2$$

$$\Rightarrow f(\vec{x}) - f(\vec{x}^*) \geq \frac{\alpha}{2}\|\vec{x}-\vec{x}^*\|^2$$

From our previous expected progress argument:

$$f(\vec{x}^{(i)}) - f(\vec{x}^{(i+1)}) \geq \frac{1}{2\beta}\|\vec{\nabla} f(x^{(i)})\|^2$$

So:

$$f(\vec{x}^{(i+1)}) - f(\vec{x}^*) \leq f(\vec{x}^{(i)}) - f(\vec{x}^*) - \frac{1}{2\beta}\left\|\vec{\nabla} f(\vec{x}^{(i)})\right\|^2$$

α-Strong convexity gives:

$$\|\vec{\nabla} f(\vec{x}^{(i)})\|^2 \geq \frac{[f(\vec{x}^{(i)}) - f(\vec{x}^*)]^2}{\|\vec{x}^* - \vec{x}^{(i)}\|^2} \geq \frac{\alpha[f(\vec{x}^{(i)}) - f(\vec{x}^*)]}{2}$$

Together, find:

$$f(\vec{x}^{(i+1)}) - f(\vec{x}^*) \leq [f(\vec{x}^{(i)}) - f(\vec{x}^*)]\left(1 - \frac{1}{2\beta}\frac{\|\vec{\nabla} f(\vec{x}^{(i)})\|^2}{f(\vec{x}^{(i)}) - f(\vec{x}^*)}\right)$$

$$\leq [f(\vec{x}^{(i)}) - f(\vec{x}^*)]\left(1 - \frac{\alpha}{4\beta}\right) \leq [f(\vec{x}^{(i)}) - f(\vec{x}^*)]\left(1 - \frac{1}{4\kappa}\right)$$

where $\kappa \equiv \frac{\beta}{\alpha}$ = condition number of f. ← smaller κ ⇒ faster convergence

Note: After $O\left(K \log \frac{[f(\vec{x}^{(0)}) - f(\vec{x}^*)]}{\varepsilon}\right)$ steps, we are within ε of optimum.

Logarithmic dependence implies excellent convergence to precise answers.

But what if our function is not strongly convex for any $\alpha > 0$?

- We could construct a new function based on f that is α-convex by adding $\alpha \|\vec{x} - \vec{x}^{(0)}\|^2$ to it (example of regularization)

$$g(\vec{x}) = f(\vec{x}) + \alpha \left\|\vec{x} - \vec{x}^{(0)}\right\|^2$$

- But now we are optimizing a different function with a possibly different optimum. So we reduce the regularization as we approach $\vec{x}^*$ by reducing $\alpha \to 0$.

- For example, we could make α a function of ε and $\|\vec{x}^* - \vec{x}^{(0)}\|^2$

L15.12