

Optimization Methods*

November 18, 2019

1 Gradient Descent Methods

1.1 Gradient Descent for Unconstrained Problems

We consider the problem of finding a minimum of a function f , hence solving

$$\min_{x \in \mathbb{R}^d} f(x)$$

where $f : \mathbb{R}^d \mapsto \mathbb{R}$ is a smooth function.

The minimum is not necessarily unique. In the general case, f might exhibit local minima, in which case the proposed algorithms are not expected to find a global minimizer of the problem. In this tour, we restrict our attention to convex function, so that the methods will converge to a global minimizer.

The simplest method is the gradient descent, that computes

$$x^{(k+1)} = x^{(k)} - \tau_k \nabla f(x^{(k)})$$

where $\tau_k > 0$ is a step size, and $\nabla f(x) \in \mathbb{R}^d$ is the gradient of f at the point x , and $x^{(0)} \in \mathbb{R}^d$ is an initial point.

In the convex case, if f is of class \mathcal{C}^2 , in order to ensure convergence, the step size should satisfy

$$0 < \tau_k < \frac{2}{\sup_x \|Hf(x)\|}$$

where $Hf(x) \in \mathbb{R}^{d \times d}$ is the Hessian of f at x and $\|\cdot\|$ is the spectral operator norm (largest eigenvalue).

The following code takes $f(x) = x^2$, computes the gradient manually and applies gradient descent to get to the solution $x = 0$.

```
tau = 2e-1;           % Step-size parameter
f = @(x)(x.^2);       % Function to minimize
fgrad = @(x)(2.*x);   % Gradient
```

*These are notes based on Gabriel Peyré's exceptional [Numerical Tours](#) and are for my own personal study.

```

x0 = -500;                % Initial guess
tol = 1e-6;               % Tolerance of the algorithm
err = 10000;              % Initial error
it = 1;                   % Iteration counter

while err > tol
    fgradx0 = fgrad(x0);
    x1(it+1) = x0-tau.*fgradx0;
    err = abs(fgradx0); % Error is the absolute value of the gradient
    if err > tol
        x0 = x1(it+1);
    end
    fprintf('New x = %3.3f \n',x0)
    it = it+1;
end

```

1.2 Gradient Descent in 2-D

Suppose we want to minimize the following quadratic form

$$f(x) = \frac{1}{2} (x_1^2 + \eta x_2^2)$$

where η controls the anisotropy and, hence, the difficulty of the problem.¹ Let us set $\eta = 10$.

The rationale for the code is the same as before. However, now I impose two stopping conditions, one for each coordinate. Note that the step-size parameter τ_k needs to be smaller than $2/\eta$. Figure

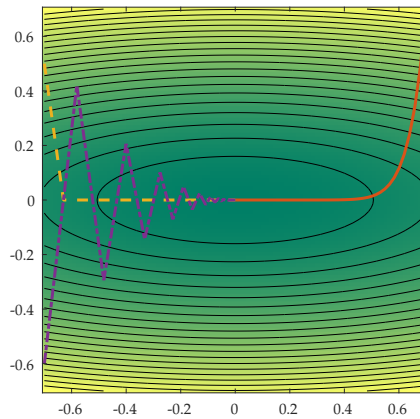


Figure 1: Visualization of Gradient Descent in 2-D for Different Values of τ_k

¹Anisotropy is the property of being directionally dependent, which implies different properties in different directions, as opposed to isotropy.