Gradient Projection Method

Will Wehner

August 11, 2020

Gradient projection can be used to solve the problem

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

where $\Omega \subset \mathbb{R}^n$ is a closed convex set. The projected gradient descent algorithm combined with armijo backtracking line-search for globalization is

Algorithm 1: Gradient Projection Method

- given $\mathbf{x}_0$
- set $\mathbf{x}_k = P_\Omega(\mathbf{x}_0)$
- Repeat $k = 0, 1, \dots, k_{\text{max}}$
  - Set search direction: $\mathbf{d}_k = P_\Omega(\mathbf{x}_k - \nabla f(\mathbf{x}_k)) - \mathbf{x}_k$
  - Set step size: α_k

    $$\begin{aligned} \alpha_k = & \max \alpha^s \\ & \text{subject to } s \in \{0, 1, 2, \dots\} \\ & f(\mathbf{x}_k + \alpha^s \mathbf{d}_k) \leq f(\mathbf{x}_k) + c\alpha^s \nabla f(\mathbf{x}_k)^T \mathbf{d}_k, \quad c \in (0, 1) \end{aligned}$$

    *requires $\mathbf{d}_k$ to be a descent direction, i.e. $\nabla f(\mathbf{x}_k)^T \mathbf{d}_k < 0$
  - new iterate $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
  - check convergence: $\|\mathbf{d}_k\|_2 \leq$ tol, or $\alpha_k \leq$ tol, or $\mathbf{d}_k$ is non descent

The gradient projection method works well when the objective function is convex and the feasible domain Ω is a set of box constraints such as bounds on the optimization variables or a polyhedron: $\Omega = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}_i^T \mathbf{x} \leq b_i, i \in \mathcal{I}\}$.

The projection operator P_Ω finds the nearest point $\mathbf{x} \in \Omega$ to a some other point $\mathbf{y}$, i.e.

$$P_\Omega(\mathbf{y}) : \quad \begin{aligned} \underset{\mathbf{x}}{\arg\min} \quad & 1/2\|\mathbf{x} - \mathbf{y}\|^2 \\ \text{subject to} \quad & \mathbf{a}_i^T \mathbf{x} \leq b_i, \quad i \in \mathcal{I} \\ & \mathbf{a}_i^T \mathbf{x} = b_i, \quad i \in \mathcal{E} \end{aligned} \tag{2}$$

An alternative description of the algorithm can be found, where instead of projecting the search direction onto the feasible domain the new iterate is projected after acceptance of step size. This version can be written as

Algorithm 2: Gradient Projection Method

- given $\mathbf{x}_0$
- set $\mathbf{x}_k = P_\Omega(\mathbf{x}_0)$
- Repeat $k = 0, 1, \ldots, k_{\text{max}}$
  - Set search direction: $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$
  - Set step size: α_k

$$\begin{aligned} \alpha_k = &\max \alpha^s \\ &\quad \text{subject to } s \in \{0, 1, 2, \ldots\} \\ &\quad f(\mathbf{x}_k + \alpha^s \mathbf{d}_k) \leq f(\mathbf{x}_k) + c\alpha^s \nabla f(\mathbf{x}_k)^T \mathbf{d}_k, \quad c \in (0, 1) \end{aligned}$$

  *requires $\mathbf{d}_k$ to be a descent direction, i.e. $\nabla f(\mathbf{x}_k)^T \mathbf{d}_k < 0$

  - new iterate $\mathbf{x}_{k+1} = P_\Omega(\mathbf{x}_k + \alpha_k \mathbf{d}_k)$
  - check convergence: $\|\mathbf{d}_k\|_2 \leq$ tol, or $\alpha_k \leq$ tol, or $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2 \leq tol$

I have found that algorithm 2 fails to converge if minimizer is on the edge of feasible set and this edge is perpendicular to the objective gradient at the minimizer (see Figure 1). The matlab code

```
matlab/test_projgrad_algo2.m
```

for a demonstration

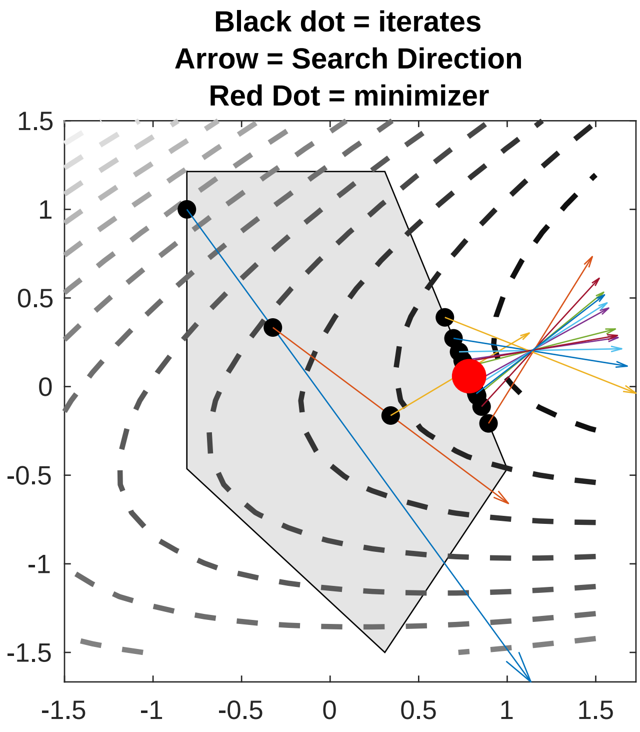

Figure 1: Algorithm 2 fail case. When the minimizer is on edge of feasible domain and the edge is perpendicular to the gradient, the search directions bounce back and forth and the algorithm fails to converge.