Lecture 5 Optimization Algorithms

Some Popular Optimization Techniques

- Gradient Descent(GD)
- Adagrad, Adam, Adadelta
- Newton’s method, Quasi-Newton’s method
- BFGS, L-BFGS
- ADMM
- Coordinate Descent
- Mirror Descent
- Projected Gradient Descent
- MCMC
- ……

Gradient Descent Algorithm

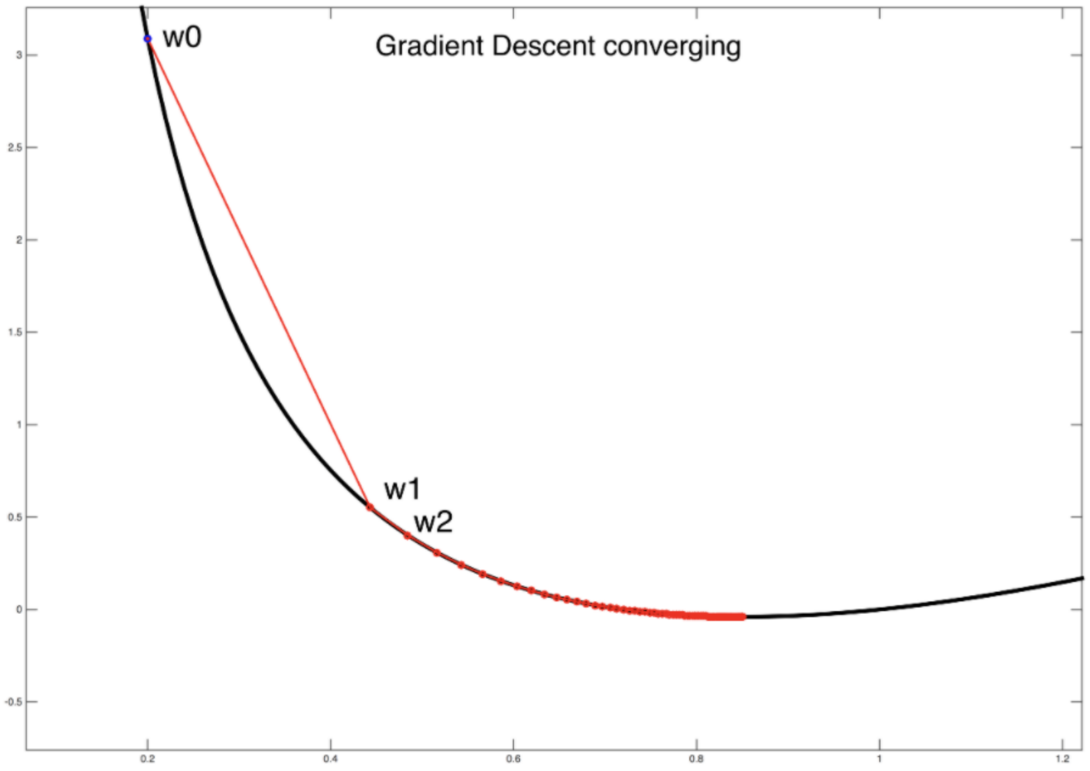
梯度下降法的过程可以表示为：

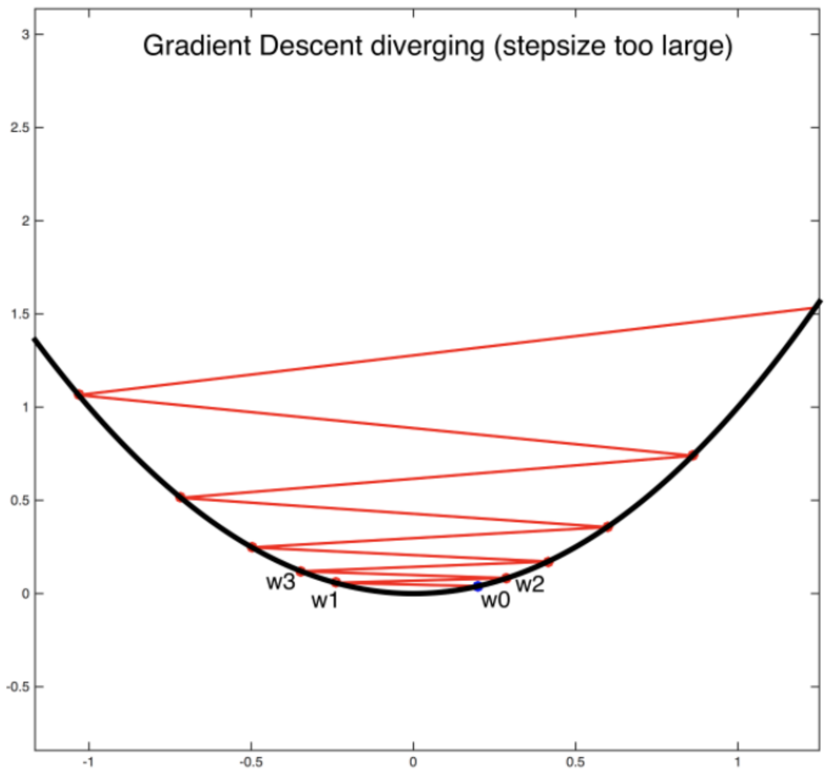
1. 选择初始值$x_0 \in R^d$, 和步长（step-size）$\eta_t > 0$
2. $for\ i = 0,1, \dots,$

$$x_{i+1} = x_i - \eta_t \nabla f(x_i)$$

Derivation of Gradient Descent

Dark art of learning rate





Gradient Descent Algorithm

Iterative Process

1. Set $x_0 \in R^d$, and learning rate（step-size）$\eta_t > 0$
2. $for\ i = 0,1, \dots,$

$$x_{i+1} = x_i - \eta_t \nabla f(x_i)$$

Question: How to analyze the complexity?

Convergence Analysis of Gradient Descent

定理

假设函数满足$L-Lipschitz$条件，并且是凸函数，设定$x^* = argminf(x)$,

那么对于步长$t \leq \frac{1}{L}$,满足

$$f(x_k) \leq f(x^*) + \frac{||x_0 - x^*||_2^2}{2\eta_t k}$$

当我们迭代k=$\frac{L||x_0-x^*||_2^2}{\epsilon}$次之后我们可以保证得到
$\epsilon - \text{approximation optimal value x}$ $(\eta_t = 1/\text{L})$

Convergence Proof

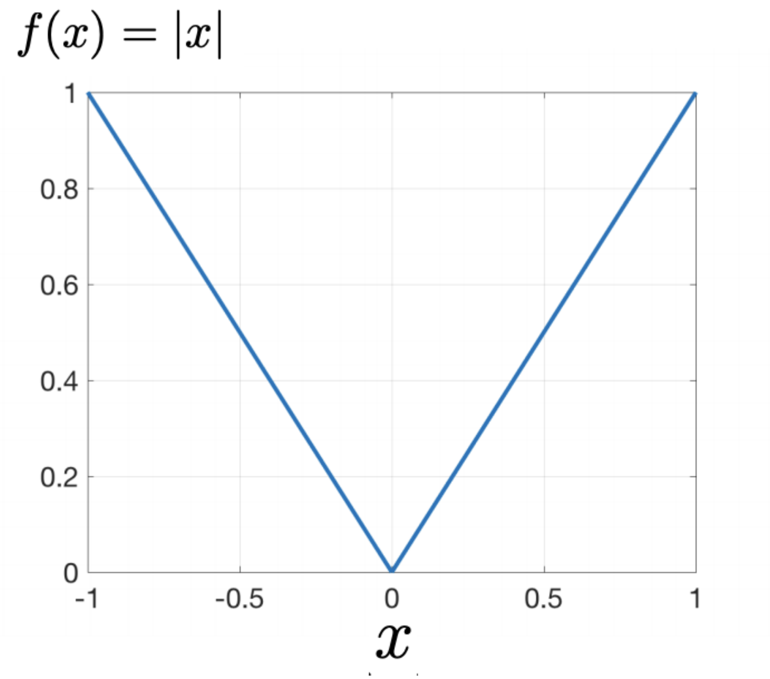
Convergence Proof

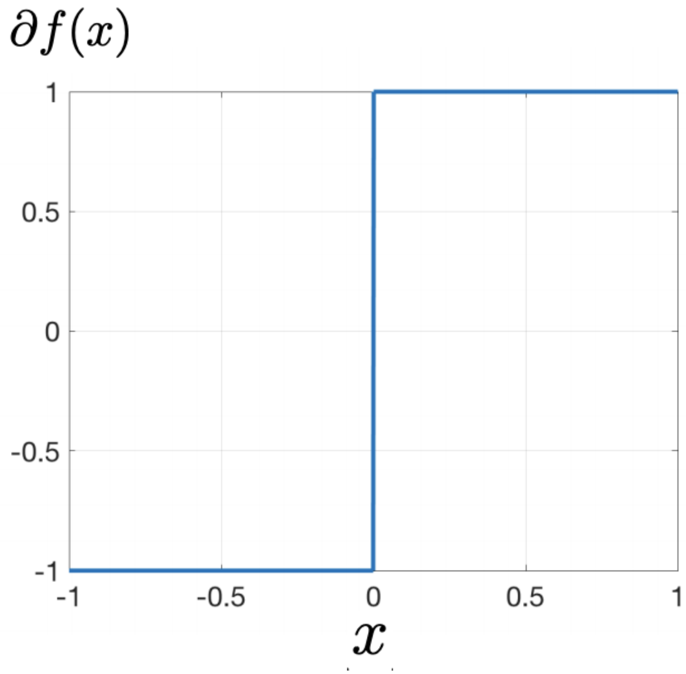
Convergence Proof

Convergence Proof

Non-differentiable Case

$$x^* = argmin\ f(x) + ||x||_1$$





$$f(x) = |x| \qquad\qquad \partial f(x) = \begin{cases} \{-1\}, & \text{if } x < 0 \\ [-1, 1], & \text{if } x = 0 \\ \{1\}, & \text{if } x > 0 \end{cases}$$

The trick of controlling learning rate

Intuition of Adagrad

Adagrad, a variant of gradient descent

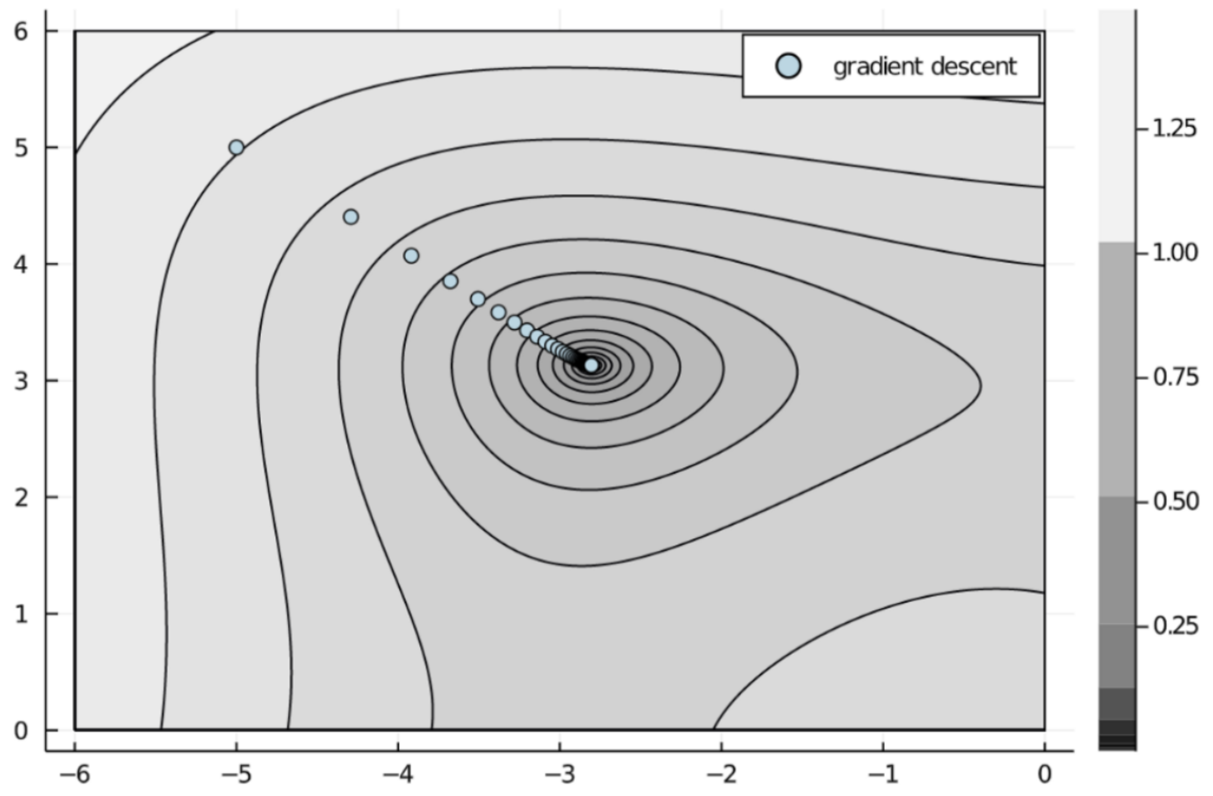
- Set the step-size adaptively for *every feature*.
- Set a small learning rate for features that have large gradients, and a large learning rate for features with small gradients

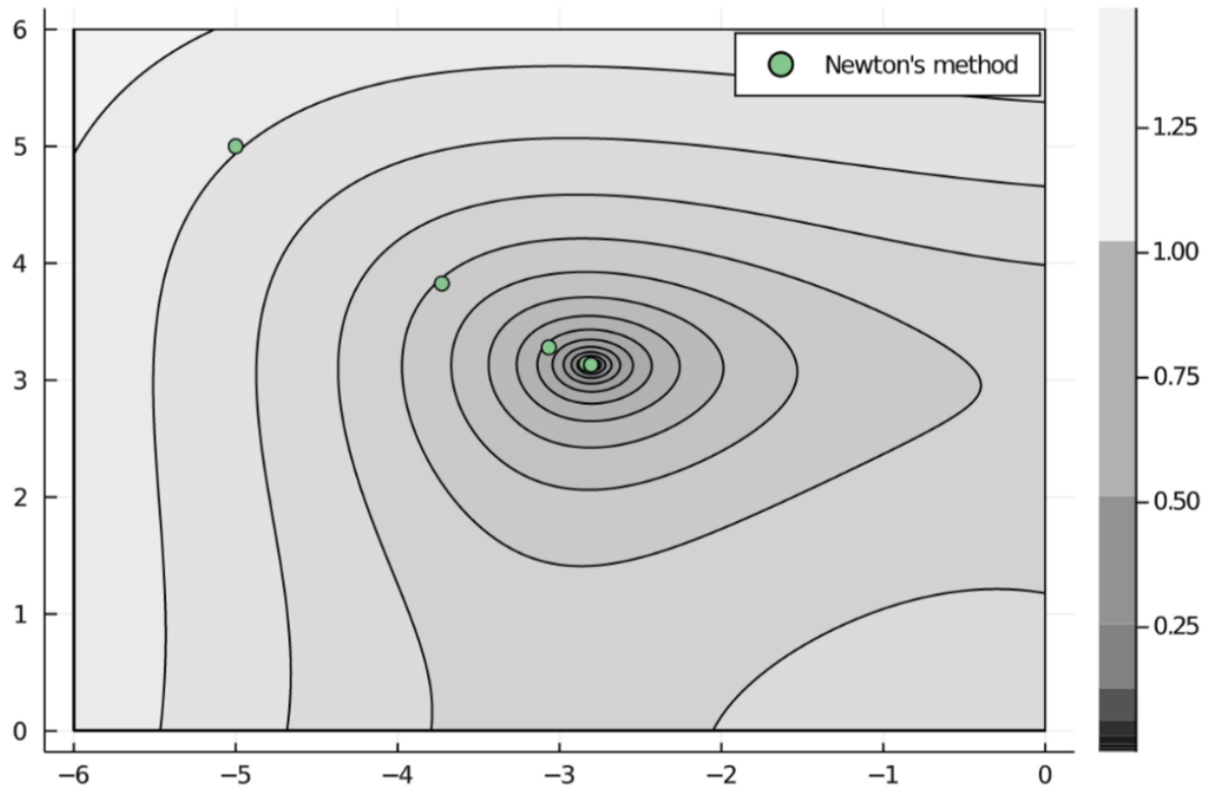
Adagrad

The problem with 1^{st} order gradient methods

Newton's method – 2nd order approximation

Convergence of Newton's Method





Advantage of Newton's Method

- Much fewer iterations compared to gradient descent
- No need to set learning rate

Looks very promising !

When Newton’s Method fail

One possible way

- Combination with gradient descent with Newton’s method

Quasi-Newton Method Derivation

Quasi-Newton Method Derivation

Quasi-Newton Method Derivation

Quasi-Newton Method Derivation

Quasi-Newton Method Derivation

Quasi-Newton Method Derivation

Projected Gradient Descent

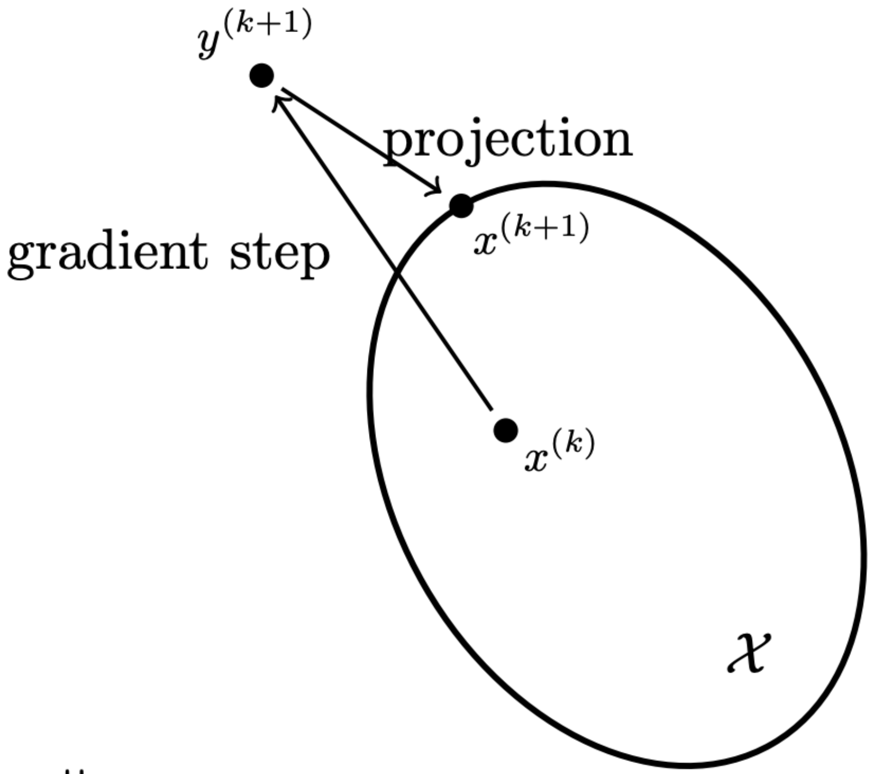
$$\min_{x} f(x)$$
$$\text{s.t. } x \in \mathcal{X}$$

Can we solve this with gradient descent?

Projected Gradient Descent

Algorithm:

- $y^{(k+1)} = x^{(k)} - t^{(k)} g^{(k)}$
  where $g^{(k)} \in \partial f(x^{(k)})$
- $x^{(k+1)} = \Pi_{\mathcal{X}}(y^{(k+1)})$



The projection operator $\Pi_{\mathcal{X}}$ onto $\mathcal{X}$:

$$\Pi_{\mathcal{X}}(x) = \min_{z \in \mathcal{X}} \|x - z\|$$

Linear regression with Nonnegative weights