Stochastic Gradient Descent from Convex Optimization Perspective

compiled by D.Gueorguiev, 3/29/2025

# Gradient Descent

Consider unconstrained smooth convex optimization

That is, is convex and differentiable with . Denote the optimal criterion value by and a solution by .

Gradient descent algorithm

choose initial point

repeat

stop when certain stopping criterion is satisfied

A grid with colored lines

Description automatically generated

A grid with colored lines and dots

Description automatically generated

At each iteration, consider the expansion

Quadratic approximation, replacing usual Hessian by

linear approximation to

proximity term to , with weight

Choose next point to minimize quadratic approximation:

A line with dots and a dotted line

Description automatically generated

Blue point is , red point is . The latter is given with

Question: How do we choose step sizes?

Simply taking for all can diverge if is too big.

A graph of a function

Description automatically generatedConsider , gradient descent after 8 steps:

Can be slow if is too small. Same example, gradient descent after 100 steps:

A graph of a graph of a stringed object

Description automatically generated with medium confidence

Convergence analysis later will give us a precise idea of “just right”

## Backtracking Line Search

One way to adaptively choose the step size is to use backtracking line search:

1 ) First fix parameters and

2 ) At each iteration, start with ,

while iter\_count < max\_iter\_count:

if :

shrink as

else:

perform gradient descent update

Simple and tends to work well in practice (further simplification: take )

A graph of a function

Description automatically generatedBacktracking interpretation

Here

Setting , backtracking picks up roughly the right step size (12 outer steps, 40 steps total).

A diagram of a dotted line

Description automatically generated with medium confidence

## Exact Line Search

We could also choose step to do the best we can along direction of negative gradient, called exact line search:

Approximations to exact line search are typically not as efficient as backtracking and it is typically not worth it.

## Convergence Analysis

Assume that f is convex and differentiable, with , and additionally that is Lipschitz continuous with constant ,

for any

or twice differentiable i.e. .

**Theorem**: Gradient descent with fixed step size satisfies

and same result holds for backtracking, with replaced by .

We say gradient descent has convergence rate . That is, it finds -suboptimal point in iterations.

## Analysis for strong convexity

**Definition**: strong convexity of means is convex for some when twice differentiable i.e for all .

**Theorem**: Assuming Lipschitz gradient as before and strong convexity one can show that gradient descent with fixed step size or with backtracking line search satisfies

where

Rate under strong convexity is , exponentially fast. That is, it finds -suboptimal point in iterations.

The gradient descent convergence is sometimes called linear convergence because the objective function value versus iteration curve looks linear on semi-log plot.

A graph of a function

Description automatically generated

//TODO: finish the Gradient Descent section

# The Sub-Gradient Method

//TODO: finish the Sub-Gradient Method

# Proximal Gradient Descent

//TODO: finish the Proximal Gradient Descent

# Stochastic Gradient Descent

//TODO: finish the Stochastic Gradient Descent

# References

[1] [Convex Optimization, Steven Boyd, Lieven Vandenberghe, 2009](https://github.com/dimitarpg13/optimization_classification_regression/blob/main/literature/books/ConvexOptimization_Boyd_2004.pdf)

[2] [Gradient Descent, Convex Optimization 10-725, CMU, slides, Ryan Tibshirani](https://github.com/dimitarpg13/statistical_learning_and_kernel_methods/blob/main/literature/articles/gradient_descent/grad-descent_Ryan_Tibshirani_slides.pdf)

[3] [Sub-Gradient Method, Convex Optimization 10-725, CMU, slides, Ryan Tibshirani](https://github.com/dimitarpg13/statistical_learning_and_kernel_methods/blob/main/literature/articles/gradient_descent/sub-gradient-method-Ryan_Tibshirani_slides.pdf)

[4] [Proximal Gradient Descent, Convex Optimization 10-725, CMU, slides, Ryan Tibshirani](https://github.com/dimitarpg13/statistical_learning_and_kernel_methods/blob/main/literature/articles/gradient_descent/proximal-gradient-descent_Ryan_Tibshirani_slides.pdf)

[5] [Stochastic Gradient Descent, Convex Optimization 10-725, CMU, slides, Ryan Tibshirani](https://github.com/dimitarpg13/statistical_learning_and_kernel_methods/blob/main/literature/articles/gradient_descent/stochastic-gradient-descent_Ryan_Tibshirani_slides.pdf)

[6] [Theory of Convex Functions, ORF 523, Princeton U., A. Ahmadi, 2015](https://github.com/dimitarpg13/optimization_classification_regression/blob/main/literature/lecture_notes/convexity/Theory_of_convex_functions_ORF523_S16_Lec7_Ahmadi_2015.pdf)

[7] Space

# Appendix

## Unconstrained Minimization

minimize (A1.1)

where is convex and twice continuously differentiable (which implies that is open).

We will assume that the problem is solvable, i.e. there exists an optimal point . We denote the optimal value as .

Since is differentiable and convex, a necessary and sufficient condition for a point to be optimal is

(A1.2)

Thus, solving the unconstrained problem (A1.1) is the same as finding solution of (A1.2) which is a set of equations in the variables . In a few special cases, we can find a solution to the problem (A1.1) by analytically solving the optimality equation (A1.2), but usually the problem must be solved by an iterative algorithm.

By this we mean an algorithm that computes a sequence of points with as . Such a sequence of points is called a *minimizing sequence* for the problem (A1.1). The algorithm is terminated when where is some specified tolerance.

### Initial point and sublevel set

The gradient descent based algorithms require a suitable starting point . The starting point must lie in , and in addition the sublevel set

(A1.3)

must be closed. This condition is satisfied for all if the function is closed, i.e. all its sublevel sets are closed. Continuous functions with are closed, so if , the initial sublevel set condition is satisfied by any . Another important class of closed functions are continuous functions with open domains, for which tends to infinity as approaches .

### Examples

Quadratic minimization and least-squares

The general convex quadratic minimization problem has the form

minimize (A1.4)

where (the set of the positive definite matrices),

## Strong Convexity

**Definition** *Strong Convexity (by Boyd and Vandenberghe)*

a function is strongly convex on iff

(A2.1)

For we have:

for some on the line segment .

By the strong convexity assumption (A1.1) the last term on the RHS is at least so we have the inequality

(A2.2)

When , we recover the basic inequality characterizing convexity; for we obtain a better lower bound on than follows from convexity alone.

We will show that the inequality (A2.2) can be used to bound , which is the suboptimality of the point , in terms of . The righthand side of (A2.2) is a convex function of (for fixed ). Setting the gradient with respect to equal to zero, we find that minimizes the RHS. Therefore, we have

.

Since this holds for any , we have

(A2.3)

This inequality shows that if the gradient is small at a point, then the point is nearly optimal. The inequality (A2.3) can also be interpreted as a condition for *suboptimality* which generalizes the optimality condition (A2.2):

(A2.4)

We can also derive a bound on , the distance between and any optimal point , in terms of :

(A2.5)

To see this, we apply (A2.2) with to obtain

where we use the Cauchy-Schwarz inequality in the second inequality.

//TODO: finish the discussion on strong convexity (Boyd and Vandenberghe)

**Theorem** *Equivalence of Strong Convexity definitions*

a function is strongly convex of modulus if and we have

or equivalently

Proof:

//TODO: finish the proof for the equivalence of strong convexity definitions

## Cauchy-Schwarz Inequality

Let and are vectors in

Then we have:

(A3.1)