

1 IMPLEMENTING GRADIENT DESCENT

Gradient descent is an iterative procedure for find a vector that minimize an objective function. Typically, the vector is our parameters and the objective function is the cost function. In each iteration, we find the gradient of the objective function evaluated at the vector and update the vector in the direction of the negative gradient.

1.1 Basic Gradient Descent

To demonstrate the gradient descent algorithm, we will begin by finding the minimum of two well defined functions with closed form derivatives shown below.

Negative Gaussian

$$f(x) = -\frac{1}{\sqrt{(2\pi)^n|\Sigma|}} \exp\left[-\frac{1}{2}(x-u)^T \Sigma^{-1}(x-u)\right] \quad (1)$$

$$\frac{\partial f(x)}{\partial x} = -f(x)\Sigma^{-1}(x-u) \quad (2)$$

Quadratic Bowl

$$f(x) = \frac{1}{2}x^T Ax - x^T b \quad (3)$$

$$\frac{\partial f(x)}{\partial x} = Ax - b \quad (4)$$

How does starting guess affect things?

Graphs of step size vs gradient norm over time.
Both gaussian and quad bowl.

Convergence criterion affects accuracy of result.

1.2 Central Difference Approximation for Gradient

1.3 Batch vs. Stochastic Gradient Descent

2 LINEAR BASIS FUNCTION REGRESSION

You can write stuff here to but don't have to.

2.1 Closed Form Solution

2.2 Gradient Descent Objective Functions

Write functions to compute SSE and derivative.

2.3 Batch vs Stochastic Gradient Descent

Compare!

2.4 Cosine Basis Functions

Shuffle.

3 RIDGE REGRESSION

3.1 Closed Form Solution

3.2 Results?

4 SPARSITY AND LASSO

4.1 Lasso