COS 324, Precept #4:
Solving Linear Systems via Gradient Descent

October 5, 2017

1 Overview

Today, we'll examine a simple application of gradient descent: iterative algorithms for approximately solving a system of linear equations. Along the way, we'll get to review some linear algebra.

2 Systems of Linear Equations

Suppose we have a system of linear equations, like

$$\begin{aligned} x_1 + x_2 &= 7, \\ 3x_1 - 2x_2 &= 1. \end{aligned}$$

In general, given an invertible matrix $A \in \mathbb{R}^{n \times n}$ and vector $b \in \mathbb{R}^n$, we want to find $x \in \mathbb{R}^n$ such that

$$Ax = b.$$

You might have heard of Gaussian or Gauss-Jordan elimination, which take linear combinations of the equations in order to isolate one variable at a time. These are pretty slow: $O(n^3)$.

The upshot is that if we wish to solve such a system up to precision ε, under some conditions, gradient-based methods will take $O(\log \frac{1}{\varepsilon})$ gradient computations of a certain function.

3 Gradient

Gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ at a point $x = (x_1, \ldots, x_n)$ is defined as $\nabla f = (\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n})$, where $\frac{\partial f}{\partial x_i}$ is the partial derivative of f w.r.t. the variable x_i. Another way to

define the gradient is a function from $\mathbb{R}^n \to \mathbb{R}^n$ which satisfies the following

$$\lim_{h \to 0} \frac{\|f(x+h) - f(x) - \nabla f^T h\|}{\|h\|} = 0$$

A simple way to find the gradient of a function is to look at $f(x+h)$ and try to express it as

$$f(x+h) = f(x) + \nabla f^T h + \text{ second order terms}$$

where second order term is typically $O(\|h\|^2)$. Let's look at an example, $f(x) = \|x\|^2 = x^T x$

$$\begin{aligned} f(x+h) &= (x+h)^T(x+h) = x^T x + x^T h + h^T x + h^T h \\ &= f(x) + 2x^T h + \|h\|^2 \\ &= f(x) + \nabla f^T h + \|h\|^2 \end{aligned}$$

So $\nabla f = 2x$. For a fixed $\|h\|$ (step size), moving in which direction will lead to the maximum increase in the function value around x? This is the same as finding h which maximizes $\nabla f^T h$ by constraining $\|h\|$ to be a constant. The direction is precisely the same direction as ∇f.

4 Convex Optimization

Recall the convex optimization framework:

$$\min_{x \in \mathcal{K}} f(x),$$

for a convex function f and convex set $\mathcal{K}$.

Let's define

$$\begin{aligned} f(x) &:= \|Ax - b\|^2 \\ &= (Ax-b)^T(Ax-b) \\ &= x^T A^T A x - 2b^T A x + b^T b. \end{aligned}$$

We can compute its gradient:

$$\nabla f(x) = 2A^T A x - 2A^T b.$$

Unsurprisingly, you can verify that this function is convex [sketch a paraboloid]. If we're at the optimum $x = A^{-1}b$, the gradient is zero. Otherwise, the gradient points in the direction of steepest ascent, so we might hope to use it to improve our solution.

Note that if A is a sparse matrix with m nonzero entries, a gradient computation costs $O(m)$. This is where we typically encounter savings in practice.

5 Gradient Descent

As we learned in class, gradient descent (in the unconstrained case $\mathcal{K} = \mathbb{R}^n$) is the following recipe for optimization:

- Start with some arbitrary x_0.
- At each step, choose $x_{t+1} \leftarrow x_t - \eta \nabla f(x_t)$.

How does gradient descent perform on this objective? It depends on the matrix A, which determines the geometry of the optimization landscape.

6 Eigenvalues and Condition Number

The *spectral theorem* states that a symmetric matrix $M \in \mathbb{R}^{n \times n}$ (such as $M = A^T A$) can be written as $M = QDQ^{-1}$, with Q orthogonal and D diagonal. In particular, the columns of Q are the eigenvectors of M, and the entries of D are their corresponding eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$.

What this means for us is that there is some choice of coordinates for which the graph of $f(x)$ looks like a standard paraboloid centered around $A^{-1}b$, stretched by a factor of λ_i in direction i.

We don't have enough time to do a complete analysis in this precept, but here's the intuition:

- When the λ_i are all close to equal (so the level sets of f look like circles), gradient descent runs really quickly.
- When this is not the case (so the level sets look like skinny ellipses), gradient descent is slower.
- When there's an eigenvalue of 0 (so $f(x)$ looks like a half-pipe), gradient descent might not converge.

7 Demos

- $M = I$.
- $M = \text{diag}(1, 9)$.