1 Some notes on solving linear systems with gradient descent

In Lecture 21, we saw an approach for solving linear systems with gradient descent. Let us briefly recall the problem setup. Our goal is to solve $Ax = b$ where A is an $n \times n$ real and invertible matrix (so we have $x \in \mathbb{R}^n$ and $b \in \mathbb{R}^n$). One way of approaching this problem is by setting up the following function $f : \mathbb{R}^n \to \mathbb{R}$:

$$f(x) \;=\; \frac{1}{2} x^T A x - x^T b \,. \tag{1}$$

We then used the following formula for the gradient:

$$\nabla f(x) \;=\; Ax - b \,. \tag{2}$$

Finally, we claimed that gradient descent solves the linear system because it converges to a point with $\nabla f(x) = 0$, which means that x solves the linear system $Ax = b$.

This route can be a great way of solving linear systems. However, there are two caveats that we did not fully explain:

- The gradient in Equation (2) is only correct if A is a *symmetric* matrix, i.e., we have $A = A^T$. In general, the gradient of the quadratic form $q(x) = \frac{1}{2} x^T A x$ is given by

  $$\nabla q(x) \;=\; \frac{1}{2} A x + \frac{1}{2} A^T x \,.$$

  Note that this is only the same as Ax if A is symmetric.

- Gradient descent can diverge with our approach unless A is a *positive semi-definite* (PSD) matrix. A symmetric matrix is PSD if and only iff we have $x^T A x \geq 0$ for all $x \in \mathbb{R}^n$.

  For a concrete example where gradient descent diverges, consider a matrix A with a negative eigenvalue. Say we want to solve $Ax = 0$ (this clearly has a trivial solution $x = 0$, but it gives a simple counterexample). Let v be the eigenvector corresponding to this negative eigenvalue. Now consider one iteration of gradient descent for this problem:

  $$x^{i+1} \;\to\; x^i - \eta \cdot A x^i$$

  where η is some fixed step size. Starting at $x^0 = v$, each iteration simply adds a multiple of v to the current iterate x^i (so x^{i+1} stays a multiple of v). So the norm of our iterates is going to grow unboundedly large! This isn't suprising: gradient descent tries to minimize a function, and if A has a negative eigenvalue, the function f is not bounded from below.

2 Gradient descent for arbitrary linear systems

At first, these two caveats might seem restrictive. However, there are actually many real problems where we want to solve a PSD linear system. In that case, our approach above is the way to go. Nevertheless, it would also be nice to solve arbitrary linear systems with gradient descent. We'll now introduce a variant of our approach above that handles the general case.

Consider the function $g : \mathbb{R}^n \to \mathbb{R}$:

$$g(x) \;=\; \frac{1}{2} \|Ax - b\|^2 \,. \tag{3}$$

We use the common notation $\|\cdot\|$ for the Euclidean norm (ℓ_2-norm) of a vector. Why does this make sense for solving a linear system? Our goal is go find an x' for which $Ax' - b$. For such an x', we have that the residual error $Ax' - b$ is the all zeros vector. So we get $g(x') = \|Ax' - b\|^2 = 0$, which is the global minimum of g (the norm of a vector is always non-negative). Conversely, if we have an x'' such that $g(x'') = 0$, we also know that x'' is a solution to our linear system.

In order to run gradient descent on g, we need the gradient. Note that we can write the Euclidean norm as a vector inner product:

$$\|z\|^2 \;=\; \sum_{i=1}^{n} z_i^2 \;=\; z^T z \;.$$

Using this identity, we rewrite g as

$$\begin{aligned} g(x) \;&=\; \frac{1}{2}(Ax-b)^T(Ax-b) \\ &=\; \frac{1}{2}x^T A^T A x - b^T A x + \frac{1}{2} b^T b \;. \end{aligned}$$

Note that $A^T A$ is now a symmetric matrix. Since we have

$$x^T A^T A x \;=\; \|Ax\|^2 \;\geq\; 0,$$

the matrix $A^T A$ is also PSD. So using the function g essentially reduces the general problem to the simpler case we have seen before!

Continuing with our gradient computation for g, we get:

$$\nabla g(x) \;=\; A^T A x - A^T b \;.$$

This is the same as the gradient for the function f above, but now with an additional factor A^T in front. Running gradient descent on g is then guaranteed to find a solution to the linear system $Ax = b$. In fact, we are actually solving a more general problem. This approach works for *arbitrary* matrices A, i.e., A does not even have to be invertible. In that case, we find an x such that the norm of $Ax - b$ is minimized. This is the *least squares* problem, which has many applications in a variety of fields.

3 Final points

We have seen that the approach in Equation (3) is more general than our original function f in Equation (1). So why would we ever use the latter formulation for solving linear systems? The reason is that gradient descent can converge faster for the function f if the matrix A is PSD. The convergence behavior of gradient descent is beyond the scope of this class. But in a nutshell, the number of gradient descent iterations we need depends on the *condition number* κ of the matrix A.[1] With our function g, we are effectively squaring the condition number. So for a PSD matrix, gradient descent can take (roughly) quadratically more steps for the function g than for the function f!

Finally, it is worth noting that there are faster variants of gradient descent for solving linear systems. One important algorithm is *conjugate gradient*, which essentially improves the condition

[1]For linear systems with large condition number, using the simple $O(n^3)$ algorithm for Gaussian elimination can actually be much faster than gradient descent. On the other hand, there are also so-called *pre-conditioning* techniques to deal with some ill-conditioned matrices efficiently.

number dependence further. Conjugate gradient can run about quadratically faster than gradient descent. But the details become more complicated. In fact, there is a well-known paper with the title “An Introduction to the Conjugate Gradient Method Without the Agonizing Pain”. We will leave it at that for this note :-)