

Lecture 8: October 26, 2015

Lecturer: Madhur Tulsiani

1 Solving systems of linear equations: Gaussian elimination

Given a system of linear equations $Ax = b$ for $A \in \mathbb{F}^{m \times n}$, $b \in \mathbb{F}^m$, recall that we can solve the system or determine that there is no solution by converting the matrix $[A \mid b]$ to a row-reduced form using elementary row operations.

Definition 1.1 A matrix $M \in \mathbb{F}^{m \times n}$ is said to be in row-reduced form if

- The first non-zero entry in each row (known as the leading entry) is 1.
- If the leading entry in row i_0 is in column j_0 , then $M_{ij} = 0$ for all $i > i_0$ and $j \leq j_0$.
- All non-zero rows occur above the zero rows.

Notice that a matrix in the row-reduced form is always upper triangular. Also, the system has no solution if and only if there is a non-zero row with a leading entry in the last column (corresponding to the entries of b). Also, if the system has a solution, then it can easily be found using back-substitution, starting from the last non-zero row.

Also, recall that an elementary row operations consist of the following (using M_i to denote the i^{th} row of M):

- Swapping the rows M_i and M_j , for some $i, j, \in [m]$
- $M_i \leftarrow c \cdot M_i$ for some $i \in [m]$, $c \in \mathbb{F} \setminus \{0\}$.
- $M_i \leftarrow M_i + c \cdot M_j$ for some $i, j \in [m]$, $c \in \mathbb{F}$.

A matrix M can always be converted to a row-reduced form using elementary row operations, which gives a general algorithm for solving a system of linear equations over any field. However, the time taken by this algorithm can be as large as $\Omega(n^3)$, which is prohibitive for large matrices. In this lecture and the next one, we will discuss methods which can take advantage of sparsity to significantly speed up the solution of linear systems.

2 Solving sparse systems of linear equations

The methods we discuss here will require analyzing distances and inner products, and thus we will work with matrices with real entries (though everything we say will extend easily to complex matrices).

Given $A \in \mathbb{R}^{m \times n}$, if we have a representation of the non-zero entries of A in “list form” i.e., a list of the non-zero entries in each row, then for any vector v , if the matrix has a total of N non-zero entries, then for any vector v , the product Av can be computed using $O(N)$ arithmetic operations. We will keep this as our base cost and try to compute a solution to $Ax = b$ using as few matrix-vector multiplications as possible.

For the purposes of the discussion below, we will assume that $A \in \mathbb{R}^{n \times n}$ is a symmetric, positive-definite matrix. This assumption is not as restrictive as it sounds, and in particular is no more restrictive than assuming that A is invertible. Given a system $A_0x = b_0$, we can always multiply both sides by A_0^T and obtain $A_0^T A_0 x = A_0^T b_0$, where the matrix $A_0^T A_0$ is now positive-definite (if A_0 is invertible). Note that $A_0^T A_0$ may not be sparse, but we can still compute $A_0^T A_0 v$ in $O(N)$ operations for any vector v using only $O(N)$ operations (we will also need the list of non-zero entries in every column for this). Taking $A = A_0^T A_0$ and $b = A_0^T b_0$ satisfies the required assumptions.

2.1 Steepest descent

Given a system $Ax = b$ with $A \succ 0$, we apply a method for minimizing the function

$$f(x) = \frac{1}{2} \cdot \langle Ax, x \rangle - \langle b, x \rangle + c$$

for some arbitrary constant $c \in \mathbb{R}$. This can be motivated by recalling that we originally had the system $A_0x = b_0$ and $Ax = b$ was obtained by multiplying both sides by A_0^T . If we consider minimizing the least square distance, we get

$$\|A_0x - b_0\|^2 = \langle A_0x, A_0x \rangle - 2 \langle b_0, A_0x \rangle + \|b_0\|^2 = \langle Ax, x \rangle - 2 \langle b, x \rangle + \|b_0\|^2.$$

Of course, scaling by a factor of 2 and changing the constant term does not change the minimizer. If x^* is the solution to the linear system, we can also re-write the above as

$$\|A_0(x - x^*)\|^2 = \langle A(x - x^*), (x - x^*) \rangle = \langle x - x^*, x - x^* \rangle_A,$$

where $\langle x, y \rangle_A$ denotes the function $\langle Ax, y \rangle$.

Exercise 2.1 Let $A \in \mathbb{R}^{n \times n}$ be a positive definite matrix. Let the function $\mu : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ be defined as $\mu(x, y) = \langle Ax, y \rangle$. Check that μ is an inner product. We will use $\langle \cdot, \cdot \rangle_A$ to this inner product.

The inner product $\langle \cdot, \cdot \rangle_A$ and the associated norm are sometimes more convenient for measuring the distance to the solution x^* since this distance actually measures the least square error in the “output” A_0x rather than the “input” x . We will need this inner product when working with the conjugate gradient method.

We will use the following algorithm for solving the linear system:

- Start with an arbitrary vector x_0 .
- At time t , update

$$x_{t+1} = x_t - \eta \cdot \nabla f(x_t) = \eta \cdot (Ax_t - b).$$

The method can also be analyzed by choosing an optimal step size η_t at each time t but we will work with the simpler variant here. Let x^* be the solution to the system $Ax = b$. We note that

$$x_{t+1} - x^* = x_t - x^* - \eta \cdot A(x_t - x^*) = (I - \eta A)(x_t - x^*).$$

By induction,

$$x_t - x^* = (I - \eta A)^t(x_0 - x^*) \Rightarrow \|x_t - x^*\|_2 \leq \|I - \eta A\|_2^t \|x_0 - x^*\|_2,$$

where we used the fact that if λ is an eigenvalue of M , then λ^t is an eigenvalue of M^t , which gives that $\|(I - \eta A)^t\|_2 = \|I - \eta A\|_2^t$. Thus, if $\|I - \eta A\|_2$ is small, we can reach a point close to the solution x^* in a small number of steps. We now choose η to minimize $\|I - \eta A\|_2$. Let $0 < \lambda_1 \leq \dots \leq \lambda_n$ denote the eigenvalues of A . Then, the eigenvalues of $I - \eta A$ are $1 - \eta\lambda_1 \geq \dots \geq 1 - \eta\lambda_n$. Thus, we have

$$\|I - \eta A\|_2 = \max\{|1 - \eta\lambda_1|, |1 - \eta\lambda_n|\}.$$

Check that this is minimized for $\eta = \frac{2}{\lambda_1 + \lambda_n}$. Plugging this, we get that

$$\|I - \eta A\|_2 = 1 - \frac{2}{\frac{\lambda_n}{\lambda_1} + 1} = 1 - \frac{2}{\kappa + 1}.$$

Here $\kappa = \lambda_n/\lambda_1$ is known as the **condition number** of the matrix A . Using this, we get that $\|x_t - x^*\| \leq \varepsilon \|x_0 - x^*\|$ after $O(\kappa \log(1/\varepsilon))$ iterations. Notice that the cost of each iteration is just $O(1)$ matrix-vector multiplications.

Exercise 2.2 Obtain a similar bound for the distance $\|x_t - x^*\|_A$ defined as $\sqrt{\langle (x - x^*), (x - x^*) \rangle_A}$.

In the next lecture, we will discuss the conjugate gradient method which can obtain a similar guarantee in $O(\sqrt{\kappa} \log(1/\varepsilon))$ iterations.