**Mathematical Toolkit** **Autumn 2016**

Lecture 10: October 27, 2016

Lecturer: Madhur Tulsiani

# 1 The conjugate gradient method

In the last lecture we saw the steepest descent or gradient descent method for finding a solution to the linear system $Ax = b$ for $A \succ 0$. The method guarantees $\|x_t - x^*\| \leq \varepsilon \cdot \|x_0 - x^*\|$ after $t = O(\kappa \cdot \log(1/\varepsilon))$ iterations, where $\kappa$ is the condition number of the matrix $A$. We will see that the conjugate gradient can obtain a similar guarantee in $O(\sqrt{\kappa} \cdot \log(1/\varepsilon))$ iterations.

For the steepest descent method, if we start from $x_0 = 0$, we get

$$x_t - x^* = (I - \eta A)(-x^*),$$

which gives $x_t = p(A) \cdot b$ for some polynomial $p$ of degree at most $t$. The conjugate gradient method just takes this idea of finding an $x$ of the form $p(A) \cdot b$ and runs with it. The method finds an $x_t = p_t(A) \cdot b$ where $p_t$ is the *best* polynomial of degree at most $t$ i.e., the polynomial which minimizes the function $\frac{1}{2} \langle Ax, x \rangle - \langle b, x \rangle$. However, the method does not explicitly work with polynomials. Instead we use the simple observation that any vector of the form $p_t(A) \cdot b$ lies in the subspace $\text{Span}\left(\{b, Ab, \ldots, A^t b\}\right)$ and the method finds the best vector in the subspace at every time $t$.

**Definition 1.1** *Let $\varphi : V \to V$ be a linear operator on a vector space $V$ and let $v \in V$ be a vector. The* Krylov subspace *of order $t$ defined by $\varphi$ and $v$ is defined as*

$$\mathcal{K}_t(\varphi, v) := \text{Span}\left(\left\{v, \varphi(v), \ldots, \varphi^{t-1}(v)\right\}\right).$$

Thus, at step $t$ of the conjugate gradient method, we find the best vector in the space $\mathcal{K}_t(A, b)$ (we will just write the subspace as $\mathcal{K}_t$ since $A$ and $b$ are fixed for the entire argument). The trick of course is to be able to do this in an iterative fashion so that we can quickly update the minimizer in the space $\mathcal{K}_{t-1}$ to the minimizer in the space $\mathcal{K}_t$. This can be done by expressing the minimizer in $\mathcal{K}_{t-1}$ in terms of a convenient orthonormal basis $\{u_0, \ldots, u_{t-1}\}$ for $\mathcal{K}_{t-1}$. It turns out that if we work with a basis which is orthonormal with respect to the inner product $\langle \cdot, \cdot \rangle_A$, at step $t$ we only need to update the component of the minimizer along the new vector $u_t$ we get to obtain a basis for $\mathcal{K}_t$.

## 1.1 The algorithm

Recall that we defined the inner product $\langle x, y \rangle_A := \langle Ax, y \rangle$ where $\langle \cdot, \cdot \rangle$ denotes the standard inner product on $\mathbb{R}^n$. As before we consider the function $\frac{1}{2}\langle Ax, x \rangle - \langle b, x \rangle$ and pick

$$x_t \; := \; \underset{x \in \mathcal{K}_t}{\arg\min} \; f(x) \,.$$

This can also be thought of as finding the closest point to $x^*$ in the space $\mathcal{K}_t$ (under the distance $\|\cdot\|_A$) since

$$\begin{aligned} f(x) \; = \; \frac{1}{2}\langle Ax, x \rangle - \langle b, x \rangle \; = \; \frac{1}{2}\langle Ax, x \rangle - \langle Ax^*, x \rangle \; &= \; \frac{1}{2}\langle x, x \rangle_A - \langle x^*, x \rangle_A \\ &= \; \frac{1}{2} \cdot \left( \|x - x^*\|_A^2 - \|x^*\|_A^2 \right) , \end{aligned}$$

which gives

$$x_t \; = \; \underset{x \in \mathcal{K}_t}{\arg\min} \; f(x) \; = \; \underset{x \in \mathcal{K}_t}{\arg\min} \, \|x - x^*\|_A \,.$$

We have already seen how to compute find the characterize the closest point in a subspace, to a given point. Let $\{u_0, \ldots, u_{t-1}\}$ be an orthonormal basis for $\mathcal{K}_t$ under the inner product $\langle \cdot, \cdot \rangle_A$. Completing this to an orthonormal basis $\{u_0, \ldots, u_{n-1}\}$ for $\mathbb{R}^n$, let $x^*$ be expressible as

$$x^* \; = \; \sum_{i=0}^{n-1} c_i \cdot u_i \; = \; \sum_{i=0}^{n-1} \langle x^*, u_i \rangle_A \cdot u_i \,.$$

Then we know that the closest point $x_t$ in $\mathcal{K}_t$, under the distance $\|\cdot\|_A$ is given by

$$x_t \; = \; \sum_{i=0}^{t-1} \langle x^*, u_i \rangle_A \cdot u_i \; = \; \sum_{i=0}^{t-1} \langle Ax^*, u_i \rangle \cdot u_i \; = \; \sum_{i=0}^{t-1} \langle b, u_i \rangle \cdot u_i$$

Note that even thhough we do not know $x^*$, we can find $x_t$ given an orthonormal basis $\{u_0, \ldots, u_{t-1}\}$, since we can compute $\langle b, u_i \rangle$ for all $u_i$. This gives the following algorithm:

- Start with $u_0 = b / \|b\|_A$ as an orthonormal basis for $\mathcal{K}_1$.
- Let $x_t = \sum_{i=0}^{t-1} \langle b, u_i \rangle \cdot u_i$ for a basis $\{u_0, \ldots, u_{t-1}\}$ orthonormal under the inner product $\langle \cdot, \cdot \rangle_A$.
- Extend $\{u_0, \ldots, u_{t-1}\}$ to a basis of $\mathcal{K}_{t+1}$ by defining

$$v_t \; = \; A^t b - \sum_{i=0}^{t-1} \left\langle A^t b, u_i \right\rangle_A \cdot u_i \quad \text{and} \quad u_t \; = \; \frac{v_t}{\sqrt{\langle v_t, v_t \rangle_A}} \,.$$

- Update $x_{t+1} \;=\; x_t + \langle b, u_t \rangle \cdot u_t$.

Notice that the basis extension step here seems to require $O(t)$ matrix-vector multiplications in the $t^{th}$ iteration and thus we will need $O(t^2)$ matrix-vector multiplications in total for $t$ iterations. This would negate the quadratic advantage we are trying to gain over steepest descent. However, in the homework you will see a way of extending the basis using only $O(1)$ matrix-vector multiplications in each step.

### 1.2 Bounding the number of iterations

Since $x_t$ lies in the subspace $\mathcal{K}_t$, we have $x_t = p(A) \cdot b$ for some polynomial $p$ of degree at most $t-1$. Thus,

$$x_t - x^* \;=\; p(A) \cdot b - x^* \;=\; p(A) \cdot A \cdot x^* - x^* \;=\; (I - p(A) \cdot A) \cdot (x_0 - x^*)\,,$$

since $x_0 = 0$. We can think of $I - p(A)A$ as a polynomial $q(A)$, where $\deg(q) \leq t$ and $q(0) = 1$. Recall from last lecture that the minimizer of $f(x)$ is the same as the minimizer of $\langle x - x^*, x - x^* \rangle_A = \|x - x^*\|_A^2$. Since $p(A)b$ is the minimizer of $f(x)$ in $\mathcal{K}_t$, we have

$$\|x_t - x^*\|_A^2 \;=\; \min_{q \in Q_t} \|q(A)(x_o - x^*)\|_A^2\,,$$

where $Q_t$ is the set of polynomials defined as

$$Q_t \;:=\; \{q \in \mathbb{R}[z] \mid \deg(q) \leq t, q(0) = 1\}\,.$$

Use the fact that if $\lambda$ is an eigenvalue of a matrix $M$, then $\lambda^t$ is an eigenvalue of $M^t$ (with the same eigenvector) to prove that the following.

**Exercise 1.2** *Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of $A$. Then for any polynomial $q$ and any $v \in \mathbb{R}^n$,*

$$\|q(A)v\|_A \;\leq\; \left(\max_i |q(\lambda_i)|\right) \cdot \|v\|_A\,.$$

Using the above, we get that

$$\|x_t - x^*\|_A \;\leq\; \left(\min_{q \in Q_t} \max_i |q(\lambda_i)|\right) \cdot \|x_0 - x^*\|_A\,.$$

Thus, the problem of bounding the norm of $x_t - x^*$ is reduced to finding a polynomial $q$ of degree at most $t$ such that $q(0) = 1$ and $q(\lambda_i)$ is small for all $i$.

**Exercise 1.3** *Verify that using $q(z) = \left(1 - \frac{2z}{\lambda_1 + \lambda_n}\right)^t$ recovers the guarantee of the steepest descent method.*

Note that the conjugate gradient method itself does not need to know anything about the optimal polynomials in the above bound. The polynomials are only used in the analysis of the bound. The following claim, which can be proved by using slightly modified Chebyshev polynomials, suffices to obtain the desired bound on the number of iterations.

**Claim 1.4** *For each $t \in \mathbb{N}$, there exists a polynomial $q_t \in Q_t$ such that*

$$|q_t(z)| \;\leq\; 2 \cdot \left(1 - \frac{2}{\sqrt{\kappa}+1}\right)^t \quad \forall z \in [\lambda_1, \lambda_n]\,.$$

We will prove the claim later using Chebyshev polynomials. However, using the claim we have that

$$\|x_t - x^*\|_A \;\leq\; \left(\min_{q \in Q_t} \max_i |q(\lambda_i)|\right) \cdot \|x_0 - x^*\|_A \;\leq\; 2 \cdot \left(1 - \frac{2}{\sqrt{\kappa}+1}\right)^t \cdot \|x_0 - x^*\|_A\,.$$

Thus, $O(\sqrt{\kappa}\log(1/\varepsilon))$ iterations suffice to ensure that $\|x_t - x^*\|_A \leq \varepsilon \cdot \|x_0 - x^*\|_A$.

### 1.3 Chebyshev polynomials

The Chebyshev polynomial of degree $t$ is given by the expression

$$P_t(z) \;=\; \frac{1}{2} \cdot \left[\left(z + \sqrt{z^2-1}\right)^t + \left(z - \sqrt{z^2-1}\right)^t\right].$$

Note that this is a polynomial since the odd powers of $\sqrt{z^2-1}$ will cancel from the two expansions. For $z \in [-1, 1]$ this can also be written as

$$P_t(z) \;=\; \cos\left(t\cos^{-1}(z)\right),$$

which shows that $P_t(z) \in [-1, 1]$ for all $z \in [-1, 1]$.

Using these polynomials, we can define the required polynomials $q_t$ as

$$q_t(z) \;=\; \frac{P_t\left(\frac{\lambda_1+\lambda_n-2z}{\lambda_n-\lambda_1}\right)}{P_t\left(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1}\right)}\,.$$

The denominator is a constant which does not depend on $z$ and the numerator is a polynomial of degree $t$ in $z$. Hence $\deg(q_t) = t$. Also, the denominator ensures that $q_t(0) = 1$. Finally, for $z \in [\lambda_1, \lambda_n]$, we have $\left|\frac{\lambda_1+\lambda_n-2z}{\lambda_n-\lambda_1}\right| \leq 1$. Hence, the numerator is in the range $[-1, 1]$ for all $z \in [\lambda_1, \lambda_n]$. This gives

$$|q_t(z)| \;\leq\; \frac{1}{P_t\left(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1}\right)} \;\leq\; 2 \cdot \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^t \;=\; 2 \cdot \left(1 - \frac{2}{\sqrt{\kappa}+1}\right)^t.$$

The last bound above can be computed directly from the first definition of the Chebyshev polynomials.

## 2 Matrices associated with graphs

Let $G = (V, E)$ be an *undirected* graph with $V = [n]$. We consider the matrices $A, D$ and $L$ associated with the graph $G$, as defined below

$$A_{ij} = \begin{cases} 1 & \text{if } \{i,j\} \in E \\ 0 & \text{otherwise} \end{cases} \qquad D_{ij} = \begin{cases} \deg(i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \qquad L = D - A.$$

The matrix $A$ is called the adjacency matrix of the graph, while $L$ is called the Laplacian matrix of the graph. We will later use these matrices to analyze properties of the underlying graph.

Note that all the above matrices are symmetric since the graph $G$ is undirected. Thus, the eigenvectors of each of these matrices form an orthonormal basis, and the corresponding eigenvalues are real. Conventionally, the eigenvalues of the Laplacian matrix are written in ascending order. Let $\lambda_1 \leq \cdots \leq \lambda_n$ be the eigenvalues of $L$. The following claim shows that $\lambda_1 \geq 0$ i.e., the Laplacian matrix is positive semidefinite.

**Proposition 2.1** *For all $x \in \mathbb{R}^n$, $\langle x, Lx \rangle = \sum_{\{i,j\} \in E} \left(x_i - x_j\right)^2$.*

Note that from the above, we also get that $\mathbf{1} \in \ker(L)$, where $\mathbf{1}$ denotes the vector $(1, \ldots, 1)$. In fact, the kernel of the Laplacian matrix corresponds directly to the connected components of $G$.

**Proposition 2.2** *Let $k$ be the number of connected components of $G$. Then $\dim(\ker(L)) = k$.*

In particular, the above says that if $\lambda_2 = 0$, then $G$ can be divided in two parts with no edges crossing between them. While this seems an uneccessarily complicated way of talking about connected components of a graph, stating this is in terms of eigenvalues has the advantage that one can talk about "robust" versions of this statement. One can then make statements of the form: if $\lambda_2$ is close to zero, then $G$ can be divided into two pieces with only a small number of edges crossing between them.

One such statement is known as *Cheeger's inequality* and we will see this later in the course. The proof of Cheeger's inequality in fact comes with an algorithm for finding a way of dividing $G$ in two pieces with small number of crossing edges. This procedure is often known as spectral partioning.