Spectral Graph Theory Lecture 12

Iterative solvers for linear equations

Daniel A. Spielman October 12, 2015

Disclaimer

These notes are not necessarily an accurate representation of what happened in class. The notes written before class say what I think I should say. I sometimes edit the notes after class to make them way what I wish I had said.

There may be small mistakes, so I recommend that you check any mathematically precise statement before using it in your own work.

These notes were last revised on October 12, 2015.

12.1 Overview

In this and the next lecture, I will discuss iterative algorithms for solving linear equations in positive semi-definite matrices.

Today's lecture will cover Richardson's first-order iterative method and the Chebyshev method. The next lecture will focus on the Conjugate Gradient method.

When studying Chebyshev's method, we will encounter Chebyshev polynomials for the second time (but of the first kind). You first saw them when we computed the characteristic polynomials of path graphs. Chebyshev polynomials are another one of the "animals in the zoo". They are one of the fundamental families of orthogonal polynomials (those satisfying three-term recurrences). These families of polynomials are the solutions to many problems, and arise naturally in many situtations. It is worth learning to identify them. We will encounter other families later in the semester, including Hermite and Laguerre polynomials.

12.2 Why iterative methods?

One is first taught to solve linear systems like

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

by *direct methods* such as Gaussian elimination, computing the inverse of $\boldsymbol{A}$, or the LU factorization. However, all of these algorithms can be very slow. This is especially true when $\boldsymbol{A}$ is sparse. Just writing down the inverse takes $O(n^2)$ space, and computing the inverse takes $O(n^3)$ time if we do

it naively. This might be OK if $\boldsymbol{A}$ is dense. But, it is very wasteful if $\boldsymbol{A}$ only has $O(n)$ non-zero entries.

In general, we prefer algorithms whose running time is proportional to the number of non-zero entries in the matrix $\boldsymbol{A}$, and which do not require much more space than that used to store $\boldsymbol{A}$.

Iterative algorithms solve linear equations *while only performing multiplications* by $\boldsymbol{A}$, and performing a few vector operations. Unlike the *direct methods* which are based on elimination, the iterative algorithms do not find exact solutions. Rather, they get closer and closer to the solution the longer they work. The advantage of these methods is that they need to store very little, and are often much faster than the direct methods. When $\boldsymbol{A}$ is symmetric, the running times of these methods are determined by the eigenvalues of $\boldsymbol{A}$.

Throughout this lecture we will assume that $\boldsymbol{A}$ is positive definite or positive semidefinite.

12.3 First-Order Richardson Iteration

To get started, we will examine a simple, but sub-optimal, iterative method, Richardson's iteration. The idea of the method is to find an iterative process that has the solution to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ as a fixed point, and which converges. We observe that if $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, then for any α,

$$\begin{aligned} \alpha\boldsymbol{A}\boldsymbol{x} &= \alpha\boldsymbol{b}, \quad \Longrightarrow \\ \boldsymbol{x} + (\alpha\boldsymbol{A} - I)\boldsymbol{x} &= \alpha\boldsymbol{b}, \quad \Longrightarrow \\ \boldsymbol{x} &= (I - \alpha\boldsymbol{A})\boldsymbol{x} + \alpha\boldsymbol{b}. \end{aligned}$$

This leads us to the following iterative process:

$$\boldsymbol{x}^t = (I - \alpha\boldsymbol{A})\boldsymbol{x}^{t-1} + \alpha\boldsymbol{b}, \tag{12.1}$$

where we will take $\boldsymbol{x}^0 = \boldsymbol{0}$. We will show that this converges if

$$I - \alpha\boldsymbol{A}$$

has norm less than 1, and that the convergence rate depends on how much the norm is less than 1. This is analogous to our analysis of random walks on graphs.

As we are assuming $\boldsymbol{A}$ is symmetric, $I - \alpha\boldsymbol{A}$ is symmetric as well, and so its norm is the maximum absolute value of its eigenvalues. Let $0 < \lambda_1 \leq \lambda_2 \ldots \leq \lambda_n$ be the eigenvalues of $\boldsymbol{A}$. Then, the eigenvalues of $I - \alpha\boldsymbol{A}$ are

$$1 - \alpha\lambda_i,$$

and the norm of $I - \alpha\boldsymbol{A}$ is

$$\max_i |1 - \alpha\lambda_i| = |\max(1 - \alpha\lambda_1, 1 - \alpha\lambda_n)|.$$

This is minimized by taking

$$\alpha = \frac{2}{\lambda_n + \lambda_1},$$

in which case the smallest and largest eigenvalues of $I - \alpha \boldsymbol{A}$ become

$$\pm \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1},$$

and the norm of $I - \alpha \boldsymbol{A}$ becomes

$$1 - \frac{2\lambda_1}{\lambda_n + \lambda_1}.$$

While we might not know $\lambda_n + \lambda_1$, a good guess is often sufficient. If we choose an $\alpha < 2/(\lambda_n + \lambda_1)$, then the norm of $I - \alpha \boldsymbol{A}$ is at most

$$1 - \alpha\lambda_1.$$

To show that $\boldsymbol{x}^t$ converges to the solution, $\boldsymbol{x}$, consider $\boldsymbol{x} - \boldsymbol{x}^t$. We have

$$\begin{aligned} \boldsymbol{x} - \boldsymbol{x}^t &= ((I - \alpha \boldsymbol{A})\boldsymbol{x} + \alpha \boldsymbol{b}) - \left((I - \alpha \boldsymbol{A})\boldsymbol{x}^{t-1} + \alpha \boldsymbol{b}\right) \\ &= (I - \alpha \boldsymbol{A})(\boldsymbol{x} - \boldsymbol{x}^{t-1}). \end{aligned}$$

So,

$$\boldsymbol{x} - \boldsymbol{x}^t = (I - \alpha \boldsymbol{A})^t (\boldsymbol{x} - \boldsymbol{x}^0) = (I - \alpha \boldsymbol{A})^t \boldsymbol{x}.$$

and

$$\begin{aligned} \left\|\boldsymbol{x} - \boldsymbol{x}^t\right\| = \left\|(I - \alpha \boldsymbol{A})^t \boldsymbol{x}\right\| &\leq \left\|(I - \alpha \boldsymbol{A})^t\right\| \|\boldsymbol{x}\| \\ &= \|(I - \alpha \boldsymbol{A})\|^t \|\boldsymbol{x}\| \\ &\leq \left(1 - \frac{2\lambda_1}{\lambda_n + \lambda_1}\right)^t \|\boldsymbol{x}\| . \\ &\leq e^{-2\lambda_1 t/(\lambda_n + \lambda_1)} \|\boldsymbol{x}\| . \end{aligned}$$

So, if we want to get a solution $\boldsymbol{x}^t$ with

$$\frac{\left\|\boldsymbol{x} - \boldsymbol{x}^t\right\|}{\|\boldsymbol{x}\|} \leq \epsilon,$$

it suffices to run for

$$\frac{\lambda_n + \lambda_1}{2\lambda_1} \ln(1/\epsilon) = \left(\frac{\lambda_n}{2\lambda_1} + \frac{1}{2}\right) \ln(1/\epsilon).$$

iterations. The term

$$\frac{\lambda_n}{\lambda_1}$$

is called the *condition number*[1] of the matrix $\boldsymbol{A}$, when $\boldsymbol{A}$ is symmetric. It is often written $\kappa(\boldsymbol{A})$, and the running time of iterative algorithms is often stated in terms of this quantity. We see that if the condition number is small, then this algorithm quickly provides an approximate solution.

[1]For general matrices, the condition number is defined to be the ratio of the largest to smallest singular value.

12.4 A polynomial approximation of the inverse

I am now going to give another interpretation of Richardson's iteration. It provides us with a polynomial in $\boldsymbol{A}$ that approximates $\boldsymbol{A}^{-1}$. In particular, the tth iterate, $\boldsymbol{x}^t$ can be expressed in the form

$$p^t(\boldsymbol{A})\boldsymbol{b},$$

where p^t is a polynomial of degree t.

We will view $p^t(\boldsymbol{A})$ as a good approximation of $\boldsymbol{A}^{-1}$ if

$$\left\|\boldsymbol{A}p^t(\boldsymbol{A}) - I\right\|$$

is small. From the formula defining Richardson's iteration (17.1), we find

$$\begin{aligned}
\boldsymbol{x}^0 &= \boldsymbol{0},\\
\boldsymbol{x}^1 &= \alpha\boldsymbol{b},\\
\boldsymbol{x}^2 &= (\boldsymbol{I} - \alpha\boldsymbol{A})\alpha\boldsymbol{b} + \alpha\boldsymbol{b},\\
\boldsymbol{x}^3 &= (\boldsymbol{I} - \alpha\boldsymbol{A})^2\alpha\boldsymbol{b} + (\boldsymbol{I} - \alpha\boldsymbol{A})\alpha\boldsymbol{b} + \alpha\boldsymbol{b}, \text{and}\\
\boldsymbol{x}^t &= \sum_{i=0}^{t}(\boldsymbol{I} - \alpha\boldsymbol{A})^i\alpha\boldsymbol{b}.
\end{aligned}$$

To get some idea of why this should be an approximation of $\boldsymbol{A}^{-1}$, consider what we get if we let the sum go to infinity. Assuming that the infinite sum converges, we have

$$\alpha\sum_{i=0}^{\infty}(\boldsymbol{I} - \alpha\boldsymbol{A})^i = \alpha\left(\boldsymbol{I} - (\boldsymbol{I} - \alpha\boldsymbol{A})\right)^{-1} = \alpha(\alpha\boldsymbol{A})^{-1} = \boldsymbol{A}^{-1}.$$

So, the Richardson iteration can be viewed as a truncation of this infinite summation.

In general, a polynomial p^t will enable us to compute a solution to precision ϵ if

$$\left\|p^t(\boldsymbol{A})\boldsymbol{b} - \boldsymbol{x}\right\| \leq \epsilon\left\|\boldsymbol{x}\right\|.$$

As $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}$, this is equivalent to

$$\left\|p^t(\boldsymbol{A})\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}\right\| \leq \epsilon\left\|\boldsymbol{x}\right\|,$$

which is equivalent to

$$\left\|\boldsymbol{A}p^t(\boldsymbol{A}) - \boldsymbol{I}\right\| \leq \epsilon$$

12.5 Better Polynomials

This leads us to the question of whether we can find better polynomial approximations to $\boldsymbol{A}^{-1}$. The reason I ask is that the answer is yes! As $\boldsymbol{A}$, $p^t(\boldsymbol{A})$ and $\boldsymbol{I}$ all commute, the matrix

$$\boldsymbol{A}p^t(\boldsymbol{A}) - \boldsymbol{I}$$

is symmetric and its norm is the maximum absolute value of its eigenvalues. So, it suffices to find a polynomial p^t such that

$$\left|\lambda_i p^t(\lambda_i) - 1\right| \leq \epsilon,$$

for all eigenvalues λ_i of $\boldsymbol{A}$.

To reformulate this, define

$$q^t(x) = 1 - xp(x).$$

Then, it suffices to find a polynomial q^t of degree $t+1$ for which

$$q^t(0) = 1, \text{ and}$$
$$\left|q^t(x)\right| \leq \epsilon, \text{ for } \lambda_1 \leq x \leq \lambda_n.$$

We will see that there are polynomials of degree

$$\ln(2/\epsilon)\left(\sqrt{\lambda_n/\lambda_1}+1\right)/2$$

that allow us to compute solutions of accuracy ϵ. In terms of the condition number of $\boldsymbol{A}$, this is a quadratic improvement over Richardson's first-order method.

Theorem 12.5.1. *For every $t \geq 1$, and $0 < \lambda_{min} \leq \lambda_{max}$, there exists a polynomial $q^t(x)$ such that*

1. $\left|q^t(x)\right| \leq \epsilon$ *, for $\lambda_{min} \leq x \leq \lambda_{min}$, and*
2. $q^t(0) = 1$,

for

$$\epsilon \leq 2(1 + 2/\sqrt{\kappa})^{-t} \leq 2e^{-2t/\sqrt{\kappa}},$$

where

$$\kappa = \frac{\lambda_{max}}{\lambda_{min}}.$$

12.6 Chebyshev Polynomials

I'd now like to explain how we find these better polynomials. The key is to transform one of the most fundamental polynomials: the Chebyshev polynomials. These polynomials are as small as possible on $[-1,1]$, and grow quickly outside this interval. We will translate the interval $[-1,1]$ to obtain the polynomials we need.

The tth Chebyshev polynomial, written T_t, may be defined as the polynomial such that

$$\cos(t\theta) = T_t(\cos(\theta)).$$

It might not be obvious that one can express $\cos(t\theta)$ as a polynomial in $\cos(\theta)$. To see that one can, recall the addition formula for cos, which gives

$$\cos(t\theta) = \cos((t-1)\theta)\cos(\theta) - \sin((t-1)\theta)\sin(\theta),$$

and

$$\cos((t-2)\theta) = \cos((t-1)\theta)\cos(\theta) + \sin((t-1)\theta)\sin(\theta).$$

Adding these two equalities together gives,

$$\cos(t\theta) + \cos((t-2)\theta) = 2\cos((t-1)\theta)\cos(\theta),$$

which we re-write as

$$\cos(t\theta) = 2\cos((t-1)\theta)\cos(\theta) - \cos((t-2)\theta).$$

This identity tells us two useful things: that $\cos(t\theta)$ can be expressed as a polynomial in $\cos(\theta)$ for integral t, and that the resulting polynomial satisfies a 3-term recurrence of the form

$$T_t(x) = 2T_{t-1}(x) - T_{t-2}(x).$$

The family of polynomials is uniquely defined once we add the initial conditions

$$T_0(x) = 1 \quad \text{and} \quad T_1(x) = 2x - 1.$$

This is very similar to, but not exactly the same as, the recurrence that is satisfied by the characteristic polynomials of path graphs.

Claim 12.6.1. *For $x \in [-1,1]$, $|T_t(x)| \leq 1$.*

Proof. For $x \in [-1,1]$, there is a θ so that $\cos(\theta) = x$. We then have $T_t(x) = \cos(t\theta)$, which must also be between -1 and 1. □

To compute the values of the Chebyshev polymomials outside $[-1,1]$, we use the hyperbolic cosine function. Recall that

$$\cosh(\theta + \phi) = \cosh(\theta)\cosh(\phi) - \sinh(\theta)\sinh(\phi),$$

and so

$$\cosh(t\theta) = 2\cosh((t-1)\theta)\cosh(\theta) - \cosh((t-2)x).$$

That is, we could also have defined the tth Chebyshev polynomial as the expansion of $\cosh(t\theta)$ in $\cosh(\theta)$. This expansion allows us to evaluate $T_t(x)$ for $|x| \geq 1$.

Recall that hyperbolic cosine maps the real line to $[1,\infty]$ and is symmetric about the origin. So, the inverse of hyperbolic cosine may be viewed as a map from $[1,\infty]$ to $[0,\infty]$. The following are the fundamental facts about hyperbolic cosine that we require:

$$\begin{aligned}
\cosh(x) &= \cos(ix) \\
\cosh(x) &= \frac{1}{2}\left(e^x + e^{-x}\right), \text{ and} \\
\operatorname{acosh}(x) &= \ln\left(x + \sqrt{x^2 - 1}\right), \text{ for } x \geq 1.
\end{aligned}$$

Claim 12.6.2. *For $\gamma > 0$,*

$$T_t(1+\gamma) \geq (1+\sqrt{2\gamma})^t/2.$$

Proof. Setting $x = 1+\gamma$, we compute

$$\begin{aligned} T_t(x) &= \frac{1}{2}\left(e^{t\,\mathrm{acosh}(x)} + e^{-t\,\mathrm{acosh}(x)}\right) \\ &\geq \frac{1}{2}\left(e^{t\,\mathrm{acosh}(x)}\right) \\ &= \frac{1}{2}(x+\sqrt{x^2-1})^t \\ &= \frac{1}{2}(1+\gamma+\sqrt{(1+\gamma)^2-1})^t \\ &= \frac{1}{2}(1+\gamma+\sqrt{2\gamma+\gamma^2})^t \\ &\geq \frac{1}{2}(1+\sqrt{2\gamma})^t. \end{aligned}$$

□

12.7 Proof of Theorem 17.6.1

We will exploit the following properties of the Chebyshev polynomials:

1. T_t has degree t.
2. $T_t(x) \in [-1,1]$, for $x \in [-1,1]$.
3. $T_t(x)$ is monotonically increasing for $x \geq 1$.
4. $T_t(1+\gamma) \geq (1+\sqrt{2\gamma})^t/2$, for $\gamma > 0$.

To express $q^t(x)$ in terms of a Chebyshev polynomial, we should map the range on which we want p to be small, $[\lambda_{min}, \lambda_{max}]$ to $[-1,1]$. We will accomplish this with the linear map:

$$l(x) \stackrel{\text{def}}{=} \frac{\lambda_{max}+\lambda_{min}-2x}{\lambda_{max}-\lambda_{min}}.$$

Note that

$$l(x) = \begin{cases} -1 & \text{if } x = \lambda_{max} \\ 1 & \text{if } x = \lambda_{min} \\ \frac{\lambda_{max}+\lambda_{min}}{\lambda_{max}-\lambda_{min}} & \text{if } x = 0. \end{cases}$$

So, to guarantee that the constant coefficient in $q^t(x)$ is one, we should set

$$q^t(x) \stackrel{\text{def}}{=} \frac{T_t(l(x))}{T_t(l(0))}.$$

We know that $|T_t(l(x))| \leq 1$ for $x \in [\lambda_{min}, \lambda_{max}]$. To find $q(x)$ for x in this range, we must compute $T_t(l(0))$. We have

$$l(0) \geq 1 + 2/\kappa(\boldsymbol{A}),$$

and so by properties 3 and 4 of Chebyshev polynomials,

$$T_t(l(0)) \geq (1 + 2/\sqrt{\kappa})^t/2.$$

Thus,

$$q(x) \leq 2(1 + 2/\sqrt{\kappa})^{-t},$$

for $x \in [\lambda_{min}, \lambda_{max}]$, and so all eigenvalues of $\boldsymbol{I} - \boldsymbol{A}q(\boldsymbol{A})$ will have absolute value at most $2(1 + 2/\sqrt{\kappa})^{-t}$.

12.8 Laplacian Systems

One might at first think that these techniques do not apply to Laplacian systems, as these are always singular. However, we can apply these techniques without change if $\boldsymbol{b}$ is in the span of $\boldsymbol{L}$. That is, if $\boldsymbol{b}$ is orthogonal to the all-1s vector and the graph is connected. In this case the eigenvalue $\lambda_1 = 0$ has no role in the analysis, and it is replaced by λ_2. One way of understanding this is to just view $\boldsymbol{L}$ as an operator acting on the space orthogonal to the all-1s vector.

By considering the example of the Laplacian of the path graph, one can show that it is impossible to do much better than the $\sqrt{\kappa}$ iteration bound that I claimed at the end of the last section. To see this, first observe that when one multiplies a vector $\boldsymbol{x}$ by $\boldsymbol{L}$, the entry $(\boldsymbol{L}\boldsymbol{x})(i)$ just depends on $\boldsymbol{x}(i-1), \boldsymbol{x}(i)$, and $\boldsymbol{x}(i+1)$. So, if we apply a polynomial of degree at most t, $\boldsymbol{x}^t(i)$ will only depend on $\boldsymbol{b}(j)$ with $i - t \leq j \leq i + t$. This tells us that we will need a polynomial of degree on the order of n to solve such a system.

On the other hand, $\sqrt{\lambda_n/\lambda_2}$ is on the order of n as well. So, we should not be able to solve the system with a polynomial whose degree is significantly less than $\sqrt{\lambda_n/\lambda_2}$.

12.9 Warning

The polynomial-based approach that I have described here only works in infinite precision arithmetic. In finite precision arithmetic one has to be more careful about how one implements these algorithms. This is why the descriptions of methods such as the Chebyshev method found in Numerical Linear Algebra textbooks are more complicated than that presented here. The algorithms that are actually used are mathematically identical in infinite precision, but they actually work. The problem with the naive implementations are the typical experience: in double-precision arithmetic the polynomial approach to Chebyshev will fail to solve linear systems in random positive definite matrices in 60 dimensions!