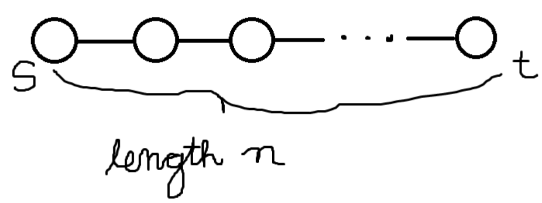
Lecture 12: Accelerated Gradient Descent and Chebyshev's Polynomial

Instructor: Lorenzo Orecchia Scribe: Sridevi Suresh

1 Outline

We have previously discussed iterative algorithms where, given a function f which is σ-strongly convex and L-smooth, we observed that after $T = O(\log\frac{D}{\varepsilon} - (\frac{L}{\sigma} + 1))$ rounds of gradient descent we would get $f(x_t) \leq f(x*) + \varepsilon$. Here, D is the diameter of the set, $x*$ is the optimal solution, and we call $\frac{L}{\sigma}$ the condition number. We will be developing a method called an accelerated gradient method which has a slightly different step and we instead have $T = O(\log\frac{D}{\varepsilon}(\sqrt{\frac{L}{\sigma}} + 1))$. It actually turns out that this is asymptotically optimal.

Let us now look further at the problem we are going to be thinking about. Suppose we have a path of length n and we want to do some electrical flow computation dealing with it:



We are further considering the Laplacian; in other words, we are trying to solve $Lx = \chi_{st}$.

$$f(x) = \min_x \frac{1}{2} x^T L x - x^T \chi_{st}$$
$$\nabla f(x) = Lx - \chi_{st}$$

We see that the only vectors that the algorithm knows is span$\{L^{(t)}\chi_{st}, L^{(t-1)}\chi_{st}, \ldots, L\chi_{st}, \chi_{st}\}$, which is called t-Krylov subspace. We can see that in order to discover this vector space $\Omega(n)$ iterations are necessary. We also notice that $\nabla^2 f = L$. Hence, our smoothness is $\lambda_n = O(1)$ and convexity is $\lambda_2 = \Omega(\frac{1}{n^2})$. Thus, roughly $\Omega(\sqrt{\frac{\lambda_n}{\lambda_2}})$ iterations are needed. This shows us that the bound on T in this case is tight; i.e we have a lower-bound.

Key-point: this bound is only valid when we do the gradient computation method.

2 Accelerated Gradient in Convex Quadratic Unconstrained Minimization

Suppose we have $f(x) = x^T Ax - b^T x$, where A is a non-singular psd matrix. It is obvious that $x* = A^{-1}b$. However, we can solve using gradient descent. Let us think of gradient descent as a polynomial. Here, we let $p(x)$ be a polynomial of x.

$$\begin{aligned}\nabla f(x) &= Ax - b \\ x_t &= \alpha x_{t-1} + B\nabla f(x_{t-1}) \\ &= \alpha x_{t-1} + BAx_{t-1} - Bb \\ &= p(A)b\end{aligned}$$

In gradient descent, we are using $p(A) = \alpha \sum_{t=0}^{k} (I - \alpha A)^t$, where α is the step-length. We want the kth polynomial $p_k(A)$ to be a good approximation to $x*$. In other words, we want $||p_k(A)b - A^{-1}b||_A$ to be small. We observe that $||p_k(A)b - A^{-1}b||_A \leq ||p_k(A) - A^{-1}||_A ||b||$. Since we are dealing with the A-norm and $||b||$ is constant, we really only care that $||p_k(A) - A^{-1}||_A$ is small. We can put A back in and write $||Ap_k(A) - I||$. This is a matrix question which we can turn into a scalar question by looking at eigenvalues individually.

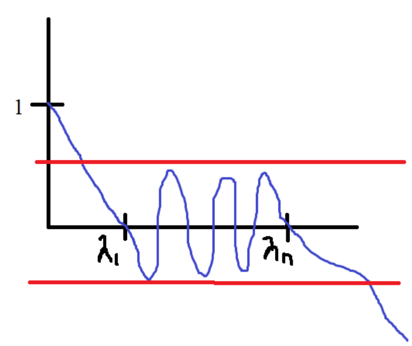
Suppose $A = \sum_{i=1}^{n} \lambda_i v_i v_i^T$, then $||Ap_k(A) - I|| = ||\sum_{i=1}^{n} (\lambda_i p_k(\lambda_i) - 1) v_i v_i^T||$. Our problem now is to find p of degree k such that $|\lambda_i p_k(\lambda_i) - 1| \leq \varepsilon$ for all λ_i eigenvalues of A. The issue is we don't know what the eigenvalues of A are. We now suppose that $\lambda_1 \leq A_i \leq \lambda_n$ and we know $lambda_1$ and λ_n. We can write out problem now to be that we need to find p_k such that $|xp_k(x) - 1| \leq \varepsilon \quad \forall x \epsilon [\lambda_1, \lambda_n]$. This is a good approximation to the inverse function.

3 Chebyshev Polynomial

Consider a polynomial $q_k(x) = 1 - xp_k(x)$. We want this polynomial to satisfy these properties:

- q_k has degree $k + 1$
- $q_k(0) = 1$
- $q_k(x) \leq \varepsilon \quad \forall x \epsilon [\lambda_1, \lambda_n]$

We can see that this polynomial looks like:



The types of polynomials we are talking about are called *Chebyshev Polynomials*. $T_k(x)$ is a Chebyshev polynomial of degree k.

Theorem Using Chebyshev, we can construct q such that $q(0) = 1$, $q(x) \leq \varepsilon \quad \forall x \epsilon [\lambda_1, \lambda_n]$, and q has degree $O(\log \frac{1}{\varepsilon} (1 + \sqrt{\frac{\lambda_n}{\lambda_1}}))$. [The algorithm which produces this is known as Chebyshev iteration.]

We have an implicit definition of our polynomial. T_t is the polynomial such that $\cos(tx) = T_t \cos(x)$. In order to understand how this works more, first, recall some trigonometric properties:

- $\cos(x + y) = \cos(x)\cos(y) + \sin(x)\sin(y)$
- $\sin(x + y) = \sin(x)\cos(y) + \cos(x)\sin(y)$
- $\sin^2(x) = 1 - \cos^2(x)$

Example Consider $\cos(2x)$. We would like to write this as some polynomial of $\cos(x)$.

$$\begin{aligned}\cos(2x) &= \cos^2(x) - \sin^2(x)\\ &= 2\cos^2(x) - 1\end{aligned}$$

From this, we can see that $T_2(x) = 2x^2 - 1$ or $T_{2t}(x) = (2T_t(x))^2 - 1$.

Suppose $T_t(x) = \cos(t \arccos(x))$ and we are considering values $[-1, 1]$. We note that $\arccos : [-1, 1] \to [-\pi, 0]$ and $\cos : [-\pi, 0] \to [-1, 1]$. Thus, we observe that if $x \ \epsilon\ [-1, 1]$, then $T_t \ \epsilon\ [-1, 1]$. Suppose further that $|x| \geq 1$; recall that arccos is not defined outside $[-1, 1]$.

In order to handle this case, we turn to hyperbolic cosine (defined over reals), $\cosh = \cos(ix) = \frac{1}{2}(e^x + e^{-x}) \approx \frac{1}{2}e^{|x|}$. Now, we have $T_t = \cosh(t \operatorname{arccosh}(x))$. We know $\operatorname{arccosh}(x) = \ln(x + \sqrt{x^2 - 1})$ for $x \geq 1$. Consequently, if $|x| \geq 1$, $T_t(x)$ is monotonically increasing. This is because we estimate $\operatorname{arccosh}(x) \approx \ln(x)$ and so $\cosh \approx e^x$ which are both monotoically increasing. Therefore, $T_t(x)$ is monotonically increasing.

Lemma $T_t(1 + \gamma) \geq \frac{(1+\sqrt{2\gamma})^t}{2}$. Let $x = 1 + \gamma$.

Proof.

$$\begin{aligned}T_t(x) &= \frac{1}{2}(e^{t \operatorname{arccosh}(x)} + e^{-t \operatorname{arccosh}(x)})\\ &\geq \frac{1}{2}(x + \sqrt{x^2 - 1})^t\\ &= \frac{1}{2}(1 + \gamma + \sqrt{(1 + \gamma)^2 - 1})^t\\ &\geq \frac{1}{2}(1 + \sqrt{2\gamma})^t\end{aligned}$$

□