CS395T: Continuous Algorithms, Part VII
Polynomial approximations

Kevin Tian

1 Motivation

In this lecture, we study the following guiding question.

Problem 1. *Given $S \subseteq \mathbb{R}$ and a function $f : S \to \mathbb{R}$, what can we say about*

$$\min_{p \in \mathcal{P}_k} \sup_{x \in S} |f(x) - p(x)|, \text{ where } \mathcal{P}_k := \{p \mid p \text{ is a degree-}k' \le k \text{ polynomial}\}?$$

Furthermore, what can we say about a p which approximately achieves the above minimum (e.g., how does it behave outside S, and can we algorithmically implement it stably)?

Note that Problem 1 is stated in terms of additive error (e.g., $|f(x) - p(x)| \le \epsilon$ for all $x \in S$), but in different situations other forms of error tolerance may be more appropriate. For example, if f is nonnegative we may ask for multiplicative error guarantees such as $|f(x) - p(x)| \le \epsilon f(x)$ for all $x \in S$. One of the most clear motivations for solving Problem 1 is the following observation.

Observation 1. *Let $p(x) = \sum_{i=0}^{k} c_i x^i$ be a degree-k polynomial, let $\mathbf{M} \in \mathbb{S}^{d \times d}$, and let $v \in \mathbb{R}^d$. We can compute $p(\mathbf{M})v$ in time $O(\mathcal{T}_{\text{mv}}(\mathbf{M}) \cdot k)$, where $p(\mathbf{M}) := \sum_{i=0}^{k} c_i \mathbf{M}^i$.*

Proof. Just plan ahead! □

In the rest of this section, we will see applications where the fundamental question, and source of ingenuity in algorithm design, is of the form in Problem 1. In all the cases, the goal is to avoid applying a matrix function $f(\mathbf{M})$ by using Observation 1 and a polynomial p instead.

1.1 Minimizing convex quadratics

Consider solving a linear regression problem of the form

$$\min_{\mathbf{x} \in \mathbb{R}^d} F(\mathbf{x}) := \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \frac{1}{2} \|\mathbf{b}\|_2^2 + \min_{\mathbf{x} \in \mathbb{R}^d} \frac{1}{2} \mathbf{x}^\top \mathbf{K} \mathbf{x} - \mathbf{x}^\top \mathbf{v}, \text{ where } \mathbf{K} := \mathbf{A}^\top \mathbf{A},\ \mathbf{v} := \mathbf{A}^\top \mathbf{b}. \quad (1)$$

This is a canonical convex optimization problem, where convexity follows from $\nabla^2 F(\mathbf{x}) = \mathbf{K} \succeq \mathbf{0}_d$. Hence, a natural approach is to use a gradient method. Observe that

$$\nabla F(\mathbf{x}) = \mathbf{K}\mathbf{x} - \mathbf{v},$$

and therefore any algorithm which produces iterates $\{\mathbf{x}_t\}_{t \ge 0}$ that are in the span of the algorithms' previous iterates and their gradients, initialized at $\mathbf{x}_0$ a multiple of $\mathbf{v}$, must have

$$\mathbf{x}_t \in \text{Span}\left\{\mathbf{v}, \mathbf{K}\mathbf{v}, \mathbf{K}^2\mathbf{v}, \ldots, \mathbf{K}^t\mathbf{v}\right\} =: \mathcal{K}_t. \quad (2)$$

We call $\mathcal{K}_t$ the order-$(t+1)$ *Krylov subspace* generated by $(\mathbf{K}, \mathbf{v})$, and such methods are correspondingly called *Krylov methods* or Krylov iteration. Notice that

$$\mathcal{K}_t = \{p(\mathbf{K})\mathbf{v} \mid p \text{ is a degree-}k' \le k \text{ polynomial}\}.$$

However, assuming $\mathbf{K}$ is full-rank for simplicity, we also have a closed-form solution $\mathbf{x}^\star$ to (1), because $\nabla f(\mathbf{x}^\star) = \mathbf{0}_d \implies \mathbf{x}^\star = \mathbf{K}^{-1}\mathbf{v}$, so it suffices to form and solve a linear system in $\mathbf{K}$ in time $O(nd^{\omega-1})$. Krylov iterations are precisely trying to approximately compute $\mathbf{x}^\star = \mathbf{K}^{-1}\mathbf{v}$ with elements of $\mathcal{K}_t$, to avoid this matrix inversion. Now suppose[1] that the eigenvalues of $\mathbf{K}$ are in $[1, \kappa]$, and consider a multiplicative variant of Problem 1, for $f(x) = x^{-1}$, $S \in [1, \kappa]$. That is, suppose

$$|f(x) - p(x)| \leq \epsilon f(x) \text{ for all } x \in [1, \kappa] \tag{3}$$

for a polynomial p of degree $k(\epsilon)$ which we can explicitly apply. Then by decomposing by eigenspaces, it is a straightforward computation that

$$\left\|p(\mathbf{K})\mathbf{v} - \mathbf{K}^{-1}\mathbf{v}\right\|_{\mathbf{K}} = \epsilon \left\|\mathbf{K}^{-1}\mathbf{v}\right\|_{\mathbf{K}},$$

which is a reasonable error metric in the high-precision regime of ϵ. The famous conjugate gradient method in numerical linear algebra solves (1) by applying the best p satisfying (3). As we derive at the end of Section 4, there exist p of degree-$\approx \sqrt{\kappa}$ which achieve this guarantee, which we know is tight from the lower bound on well-conditioned optimization in Section 4, Part II.

This is an example of a more general strategy known as the *Lanczos method* or Lanczos iteration, which is a generic reduction from approximately applying a matrix function to some explicit $O(t)$-dimensional computations associated with an order-t Krylov subspace. We informally summarize its guarantees here, based on a recent analysis in finite-precision arithmetic.

Theorem 1 ([MMS18], informal)**.** *There is an algorithm (the Lanczos method) which takes as input* $\mathbf{A} \in \mathbb{S}^{d \times d}$ *with* $\|\mathbf{A}\|_{\text{op}} = \text{poly}(d)$*,* $k \in \mathbb{N}$*,* $\epsilon \in (0, 1)$ *and a function* f *whose domain contains* $[\lambda_{\min}(\mathbf{A}) - \epsilon, \lambda_{\max}(\mathbf{A}) + \epsilon]$ *and whose range is* $\text{poly}(d)$*. The algorithm takes time* $\widetilde{O}(\mathcal{T}_{\text{mv}}(\mathbf{A}) \cdot k)$*, uses* $O(\log(\frac{d}{\epsilon}))$ *bits of precision, and returns* $\mathbf{y}$ *satisfying* $\|f(\mathbf{A})\mathbf{x} - \mathbf{y}\|_2 \leq O(k \cdot \delta_k + \epsilon) \|\mathbf{x}\|_2$*, where*

$$\delta_k := \min_{p \in \mathcal{P}_{k-1}} \left(\max_{x \in [\lambda_{\min}(\mathbf{A}) - \epsilon, \lambda_{\max}(\mathbf{A}) + \epsilon]} |p(x) - f(x)| \right).$$

Theorem 1 is a bit of a mouthful, but roughly it says the Lanczos method returns y which approximates $f(\mathbf{A})\mathbf{x}$ as well as $p(\mathbf{A})\mathbf{x}$, where p is the best approximation to f in $\mathcal{P}_{k-1}$ (up to a k factor in the accuracy, and some ϵ factors which the bit complexity depends polylogarithmically on). Indeed, Theorem 1 competes up to an $O(k)$ factor with the "gold standard" of Krylov methods,

$$\|f(\mathbf{A})\mathbf{x} - p(\mathbf{A})\mathbf{x}\|_2 \leq \|f(\mathbf{A}) - p(\mathbf{A})\|_{\text{op}} \|\mathbf{x}\|_2 \leq \delta_k \|\mathbf{x}\|_2.$$

Here, we used the definition of δ_k (for any $\epsilon > 0$) to bound $\|f(\mathbf{A}) - p(\mathbf{A})\|_{\text{op}} \leq \delta_k$ by considering each eigenvalue separately. So, we can bound the performance of the Lanczos method simply by proving existence of $p \in \mathcal{P}_{k-1}$ which approximates f well, without calculating p. This alleviates much of the computational burden associated with approximate solutions to Problem 1. For the rest of the lecture, we use Theorem 1 as an excuse to not discuss the computation of the approximating polynomials we construct. In many applications the polynomial we wish to apply is explicit, which lets us avoid the tedium of Theorem 1 (and shave off low-order polylogarithmic runtime terms).

For convenience, we give a sketch of the proof of Theorem 1 under exact arithmetic in Section 5.

1.2 Principal component analysis

Consider the problem of approximating the top right singular vector of a matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$, or the top eigenvector of $\mathbf{K} := \mathbf{A}^\top \mathbf{A}$, i.e., 1-PCA. The most classical method for doing so is the *power method*, which samples a random Gaussian vector $\mathbf{g} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$ and applies $\mathbf{K}^m \mathbf{g}$ for some appropriately large m, and then normalizes the result. We will see a correctness analysis of the power method in Part XI. For now, note that Observation 1 gives us the natural strategy of applying a lower-degree polynomial q, such that $q(\mathbf{K}) \approx \mathbf{K}^m$, as done in [MM15]. As we will see in Lemma 1, such polynomials q of degree $\approx \sqrt{m}$ exist, and are quite useful in algorithm design. We note that [MM15] generalizes this result to k-PCA, using ideas inspired by the Lanczos method.

1.3 Principal component projection

Consider a denoising task, where we wish to preprocess noisy data $\mathbf{X} \in \mathbb{R}^{n \times d}$ by projecting it onto a low-dimensional subspace, with projection matrix $\mathbf{\Pi} \in \mathbb{R}^{d \times d}$. In natural statistical models,

[1] Suppose instead that we knew the eigenvalues are in some range $[\mu, L]$ with $\kappa := \frac{L}{\mu}$. We use $\mathbf{K} \leftarrow \frac{1}{\mu}\mathbf{K}$ instead.

discussed in a later lecture, a reasonable choice is $\mathbf{\Pi} = \mathbf{V}\mathbf{V}^\top$ where $\mathbf{V} \in \mathbb{R}^{d\times k}$ is the top-k right singular vectors of $\mathbf{X}$. Suppose that we know $\mathbf{K} := \mathbf{X}^\top\mathbf{X}$ with eigenvalues $\{\lambda_i\}_{i\in[d]}$ satisfies $(1-\gamma)\lambda_k \ge \lambda \ge (1+\gamma)\lambda_{k+1}$. In other words, there is a multiplicative *gap* of width $\approx \gamma$ between λ_k and λ_{k+1}. Here, the goal is to apply $\mathbf{\Pi} = f(\mathbf{K})$, where $f : [0,\infty) \to \mathbb{R}$ satisfies

$$f(x) = \begin{cases} 1 & x \ge \lambda_k \\ 0 & x \le \lambda_{k+1} \end{cases}.$$

Now consider the solution to a ridge regression problem,

$$\mathbf{x}^\star_\lambda := (\mathbf{K} + \lambda\mathbf{I}_d)^{-1}\,\mathbf{K}\mathbf{x}.$$

The main benefit to using $\mathbf{x}^\star_\lambda$ in place of $\mathbf{\Pi}\mathbf{x}$ is computational, as regularizing by $\lambda\mathbf{I}_d$ improves problem conditioning, compared to computing $\mathbf{\Pi}$ which can be expensive. Furthermore, ridge regression can be thought of as a soft proxy for explicit projection. Notice that this solution can also be rewritten as $\mathbf{x}^\star_\lambda = f(\mathbf{K})\mathbf{x}$, where

$$f(a) = \frac{a}{a+\lambda} \text{ for all } a \ge 0.$$

Additionally, f takes the range $[0,\infty)$ to $[0,1]$, and in particular $f(\lambda_k) = \frac{1}{2} + \Omega(\gamma)$ and $f(\lambda_{k+1}) = \frac{1}{2} - \Omega(\gamma)$, by assumption. This shows that $\mathbf{K}' = (\mathbf{K} + \lambda\mathbf{I}_d)^{-1}\mathbf{K}$, our initial approximation to $\mathbf{\Pi}$, can be significantly boosted in approximation quality. Indeed, it suffices to compute $p(\mathbf{K}')\mathbf{x}$, where p is a polynomial which sends $[\frac{1}{2} + \Omega(\gamma), 1]$ to values near 1, and $[0, \frac{1}{2} - \Omega(\gamma)]$ to values near 0 (with arbitrary behavior in the gap around $\frac{1}{2}$). Interestingly, [FMMS16] shows that in various settings, this gives a principal component projection algorithm which does not need to explicitly form an approximation to $\mathbf{V}$, bypassing principal component analysis.

1.4 Matrix multiplicative weights

We now briefly introduce the central topic of next lecture. The matrix multiplicative weights algorithm is a ubiquitous tool in modern algorithm design, as an abstraction of semidefinite programming. It also captures regret minimization over $\mathcal{Y} \subseteq \mathbb{S}^{d\times d}$, the set of positive semidefinite trace-1 matrices, which is often helpful in quantum computing. The algorithm maintains an iterate

$$\mathbf{Y} \propto \exp(\mathbf{S}),$$

for $\mathbf{S}$ a running sum of "gain matrices" given to the algorithm, where the constant of proportionality ensures $\mathrm{Tr}(\mathbf{Y}) = 1$. Here, the key computational challenge is simulating approximate matrix-vector access through $\exp(\mathbf{S})$, when $\mathbf{S} \in \mathbb{S}^{d\times d}$ is explicit. The naïve strategy applies eigendecomposition and practical implementations take $\Omega(d^3)$ time as a result. Instead, one can apply Problem 1 and ask for a polynomial approximation to exp. Due to how frequently the matrix multiplicative weights algorithm appears in applications, approximating exp is a central theme of the lecture.

1.5 Jackson's theorem

Finally, we mention one application in statistics (rather than linear algebra) the author of these notes is particularly fond of.[2] The earth mover's (a.k.a. 1-Wasserstein) distance between two distributions P and Q, supported on a set S with distance function $d : S \times S \to \mathbb{R}_{\ge 0}$, is defined as:

$$\mathrm{EMD}(P,Q) := \inf_{\gamma\in\Gamma(P,Q)} \int d(x,x')\mathrm{d}\gamma(x,x'). \tag{4}$$

Here, $\Gamma(P,Q)$ is the set of *couplings* of P, Q, which are joint distributions supported on $S \times S$ whose marginals agree with P and Q. It turns out that by viewing EMD as a linear program (with marginal constraints on the decision variable γ), and taking a dual, we arrive at another characterization of EMD. We state the following well-known result without proof.

Fact 1. *For any distributions P, Q supported on a set S with distance function $d : S \times S \to \mathbb{R}_{\ge 0}$ such that $d(x,y) = \|x-y\|$ for some norm $\|\cdot\|$, following the definition (4),*

$$\mathrm{EMD}(P,Q) = \sup_{g\in\mathcal{G}} \int g(x)(P(x)-Q(x))\mathrm{d}x, \text{ where } \mathcal{G} := \{g : S \to \mathbb{R} \mid g \text{ is 1-Lipschitz in } \|\cdot\|\}.$$

[2] This was the subject of my first paper in graduate school.

This is an opportunity to introduce a powerful tool in approximation theory (proof in e.g., [vP15]).

Theorem 2 (Jackson)**.** *Suppose $f : S \to \mathbb{R}$ is k-times differentiable, and $|f^{(k)}| \le L$ everywhere in S. There are constants c, C such that if $k \le cd$, there is a degree-d polynomial p such that*

$$\sup_{x \in S} |f(x) - p(x)| \le L \cdot \left(\frac{C}{d}\right)^k .$$

Note that Theorem 2 states that L-Lipschitz functions admit approximations by degree-d polynomials with additive approximation quality $O(\frac{L}{d})$. Combined with Fact 1, Theorem 2 proves that any two distributions P, Q which agree on their first $O(\frac{1}{\epsilon})$ moments must have $\text{EMD}(P, Q) \le \epsilon$. This is because $\text{EMD}(P, Q) = \int g(x)(P(x) - Q(x))\text{d}x = \int (g(x) - p(x))(P(x) - Q(x))\text{d}x \le \epsilon$.

Now consider the problem of learning a distribution P, say supported on $[0, 1]$, in earth mover's distance, up to error ϵ. It suffices to learn the first $O(\frac{1}{\epsilon})$ moments of P, i.e., $\mathbb{E}[P^i]$ for $i \in [O(\frac{1}{\epsilon})]$, from samples. Of course, we cannot exactly learn these moments due to randomness in the sampling process, leading to the development of *error-tolerant* variants of Theorem 2 which allow for small magnitudes of $\mathbb{E}[P^i - Q^i]$. We can then produce any Q which matches our estimated moments and attain an EMD guarantee. This strategy was followed by [KV17] to learn the spectrum of a graph, and then by [TKV17] to learn a population of binomial random variables.

2 Chebyshev polynomials

Chebyshev polynomials (of the first kind) are perhaps the most ubiquitous tool in algorithmic polynomial approximation. We begin with a definition. Let $T_0(x) := 1$, $T_1(x) := x$, and let

$$T_k(x) := 2xT_{k-1}(x) - T_{k-2}(x) \tag{5}$$

be the degree-k Chebyshev polynomial, recursively defined for $k \ge 2$. The Chebyshev polynomials have a particularly interesting interpretation on the range $S := [-1, 1]$, which is the focus of the rest of the lecture.[3] Indeed, the identity $2\cos(\theta)\cos((k-1)\theta) = \cos((k-2)\theta) + \cos(k\theta)$ shows

$$T_k(\cos(\theta)) = \cos(k\theta) \text{ for all } \theta \in [-\pi, \pi]. \tag{6}$$

As a result of (6), $|T_k(x)| \le 1$ for all $x \in [-1, 1]$. Moreover, Chebyshev polynomials form an *orthogonal basis* with respect to an appropriate measure on $[-1, 1]$, in that

$$\int_{-1}^{1} T_k(x)T_{k'}(x)\frac{1}{\sqrt{1-x^2}}\text{d}x = \begin{cases} 1 & k = k' \\ 0 & k \ne k' \end{cases}.$$

Hence, any Lipschitz function $f : [-1, 1] \to \mathbb{R}$ has a unique decomposition

$$f(x) = \sum_{i=0}^{\infty} a_i T_i(x) \tag{7}$$

in the Chebyshev basis. We even have an explicit formula for the Chebyshev coefficients,

$$a_i = \frac{1}{\pi\iota}\int_{|z|=1} z^{-(i+1)} f\left(\frac{1}{2}(z + z^{-1})\right)\text{d}z. \tag{8}$$

When studying polynomial approximations over $[-1, 1]$, the orthogonality and boundedness properties of Chebyshev polynomials over this interval are extremely useful. For example, simply truncating the decomposition (7) up to order k lets us upper bound uniform approximation quality in terms of the sizes of Chebyshev coefficients. As we explain in Section 4, these coefficients decay exponentially in standard applications. Interestingly, in some cases this Chebyshev truncation strategy is provably optimal up to a constant factor; see [AA22] for a proof in the case $f(x) = \exp(Cx)$ for $C \in \mathbb{R}$. We also briefly mention that Chebyshev polynomials have long-been studied due to their extremal properties, such as the following.

[3] This is without loss of generality when S is an interval; additive approximations to $f(\cdot)$ on $[a, b]$ are the same as additive approximations to $f(\frac{b-a}{2} \cdot + \frac{b+a}{2})$ on $[-1, 1]$. See [TT24], Section 3.3 for an example of machinery which extends polynomial approximations to functions defined on the union of intervals.

1. Chebyshev polynomials (scaled appropriately) solve Problem 1 optimally when $S = [-1, 1]$ and $f(x) = x^{k+1}$, i.e., they are the best degree-k approximation to x^{k+1}.
2. Let p be a degree-k polynomial such that $p([-1, 1]) \subseteq [-1, 1]$. Then for any y with $|y| \geq 1$, we have $|p(y)| \leq |T_k(y)|$. This implies T_k uniformly grows faster outside $[-1, 1]$ than *any polynomial* p of equal degree which enjoys a similar bound $|p(x)| \leq 1$ within $[-1, 1]$.
3. Similarly, for odd k, among all degree-k polynomials p such that $p([-1, 1]) \subseteq [-1, 1]$, $T_k(x)$ has the largest derivative at 0.

There are entire books written about the amazing properties of Chebyshev polynomials, e.g., [Tre19]. Even the mere fact that they admit the simple recursive definition (5) has significant consequences (and is the basis for momentum methods in optimization, as well as the Lanczos method in Theorem 1). To give the reader a sense of how to use Chebyshev polynomials, we go in depth into a few of their most important applications in Sections 3 and 4.

3 Approximating monomials

The first meta-strategy we will see in constructing polynomial approximations is based on the following cornerstone result, which gives a low-degree approximation of the monomial x^k. Roughly speaking, the meta-strategy is to first truncate the Taylor series of a function of interest, and then apply Lemma 1 to further reduce to the degree of each monomial in the truncated Taylor series.

Lemma 1. *Let $k, m \in \mathbb{N}$. There is a degree-k polynomial p satisfying*

$$\sup_{x \in [-1,1]} |p(x) - x^m| \leq 2 \exp\left(-\frac{k^2}{2m}\right).$$

Proof. We reproduce an extremely elegant proof due to [SV14]. Let $\{Y_i\}_{i \in \mathbb{N}}$ be i.i.d. Rademacher random variables, and let $D_m := \sum_{i \in [m]} Y_i$. We claim that by induction on m, $\mathbb{E}[T_{D_m}(x)] = x^m$, where $T_k(x) := T_{|k|}(x)$ for $k < 0$.[4] To see this,

$$x^{m+1} = x\mathbb{E}\left[T_{D_m}(x)\right] = \mathbb{E}\left[xT_{D_m}(x)\right] = \mathbb{E}\left[\frac{T_{D_m+1}(x) + T_{D_m-1}(x)}{2}\right] = \mathbb{E}\left[T_{D_{m+1}}(x)\right],$$

where the first equality used the inductive hypothesis, and the third used (5). Now define

$$p(x) := \mathbb{E}\left[T_{D_m}(x) \cdot \mathbb{I}_{|D_m| \leq k}\right],$$

where $\mathbb{I}_{|D_m| \leq k}$ indicates the event $|D_m| \leq k$. Clearly $p(x)$ has degree at most k by definition. To prove the claimed approximation bound, we have for $x \in [-1, 1]$,

$$|p(x) - x^m| = \left|\mathbb{E}\left[T_{D_m}(x) \cdot (1 - \mathbb{I}_{|D_m| \leq k})\right]\right| \leq \left|\mathbb{E}\left[\mathbb{I}_{|D_m| > k}\right]\right| = \Pr\left[|D_m| > k\right],$$

since Chebyshev polynomials are bounded in ±1 over $[-1, 1]$. The conclusion follows from Hoeffding's inequality (Fact 2, Lemma 1, and Theorem 1, Part VI), i.e., $\Pr[|D_m| > k] \leq 2\exp(-\frac{k^2}{2m})$. □

Taking k larger than $\sqrt{m}$ by logarithmic factors in Lemma 1 already achieves highly-accurate approximations to x^m, which can improve the degree of polynomial approximations termwise. To illustrate, we carry out our meta-strategy when $f(x) = \exp(-Cx)$ and $S = [0, 1]$. In this example, truncating the Taylor series of f at degree k gives accurate approximations when $k \gg C$, as

$$f(x) = \sum_{i=0}^{k} \frac{C^i}{i!} x^i + \sum_{i > k} \frac{C^i}{i!} x^i,$$

and $i!$ grows much faster than C^i when $i \gg C$. Using Lemma 1 to further approximate each monomial in the Taylor expansion achieves a quadratic improvement over this strategy.

[4] It is straightforward to verify that (5) continues to hold under this definition.

Lemma 2. *Let $C > 0$ and $\delta \in (0,1)$. There is a degree-k polynomial p satisfying*

$$\sup_{x\in[0,1]} |\exp(-Cx) - p(x)| \le \delta,\ k = O\left(\sqrt{C \log \frac{1}{\delta}} + \log \frac{1}{\delta}\right).$$

Proof. We begin by shifting the range to $[-1,1]$, defining

$$g(y) := \exp\left(-C\left(\frac{y+1}{2}\right)\right) = \exp\left(-\lambda - \lambda y\right),\ \text{where } \lambda := \frac{C}{2}.$$

Providing δ-approximations to g over $[-1,1]$ yields the conclusion by identifying $x = \frac{y+1}{2}$, which does not affect degrees. Next, we truncate the Taylor series of g at degree $t \in \mathbb{N}$ to be specified:

$$g(y) = \exp\left(-\lambda\right) \sum_{i=0}^{t} \frac{(-\lambda)^i}{i!} y^i + \exp\left(-\lambda\right) \sum_{i>t} \frac{(-\lambda)^i}{i!} y^i.$$

Letting $p_{k,m}$ be the degree-k approximation of x^m from Lemma 1, we define our approximation

$$q(y) := \exp(-\lambda) \sum_{i=0}^{t} \frac{(-\lambda)^i}{i!} p_{k,i}(y).$$

Observe that for $y \in [-1,1]$, by Lemma 1, we have

$$|g(y) - q(y)| \le \underbrace{\exp(-\lambda)\left(\sum_{i=0}^{t} \frac{\lambda^i}{i!} \cdot 2\exp\left(-\frac{k^2}{2i}\right)\right)}_{:=T_1} + \underbrace{\exp\left(-\lambda\right) \sum_{i>t} \frac{\lambda^i}{i!}}_{:=T_2}.$$

Since $T_1 \le 2\exp(-\frac{k^2}{2t})$, choosing $k \gtrsim \sqrt{t \log \frac{1}{\delta}}$ ensures $T_1 \le \frac{\delta}{2}$. Similarly, it is straightforward to check that $t \gtrsim \max(\lambda, \log \frac{1}{\delta})$ suffices for $T_2 \le \frac{\delta}{2}$. Combining with $\lambda \approx C$ gives the claim. □

Remark 1. *The setting of Lemma 2 appears somewhat strange, as it concerns $x \in [0,1]$ rather than $x \in [-1,1]$. In applications discussed in the next lecture, this is not prohibitive as we can simply scale x and C appropriately (e.g., $x \leftarrow 1 + x$ and $C \leftarrow \frac{C}{2}$ moves the range $[-1,1]$ to $[0,1]$). More importantly, the optimal polynomial approximation to $\exp(-Cx)$ on the range $[-1,0]$ actually has degree-$\Omega(C)$ [AA22], nullifying the $\sqrt{C}$ savings due to the strategy implied by Lemma 1.*

Remark 2. *Our strategy in establishing Lemma 2 begs the question, why not apply the monomial approximation again (and, say, obtain a polynomial approximation of degree $\approx \sqrt[4]{C}$)? The issue is that coefficients of Chebyshev polynomials rapidly blow up (T_k has leading coefficient 2^{k-1}), and we incur multiplicative factors in the approximation quality based on the sizes of these coefficients. For example, approximating $2^k x^k$ to additive error δ is the same as approximating x^k to the extremely small additive error $\delta \cdot 2^{-k}$, immediately washing out the gains of applying Lemma 1 again.*

4 Trefethen's theorem

In this section, we give an analysis of the Chebyshev truncation strategy from Section 2. Recall that, following the notation (7), this strategy proposes to use $p(x) = \sum_{i=0}^{k} a_i T_i(x)$ as our polynomial approximation. This is useful when the $\{a_i\}_{i\ge 0}$ decay quickly, as seen in the following.

Lemma 3. *Let $f(x)$ have decomposition (7) over $[-1,1]$, and let $p(x) := \sum_{i=0}^{k} a_i T_i(x)$. Then*

$$\sup_{x\in[-1,1]} |f(x) - p(x)| \le \sum_{i>k} |a_i|.$$

Proof. It suffices to apply the triangle inequality and $|T_i(x)| \le 1$ for $x \in [-1,1]$. □

For example, if we can estimate $|a_i| \leq \exp(-\epsilon i)$ for sufficiently large i, Lemma 3 shows that a degree-$\approx \epsilon^{-1}$ polynomial approximation provides strong uniform approximations. Amazingly, for all functions f which are analytic over a small extension of the interval $[-1,1]$ in the complex plane, the $\{a_i\}_{i\geq 0}$ do indeed decay exponentially, as captured by the following result.

Theorem 3 ([Tre19])**.** *Let f be analytically continuable to the interior of*

$$E_\rho := \left\{\frac{1}{2}(z+z^{-1}) \mid |z| = \rho\right\}, \text{ the Bernstein ellipse of radius } \rho,$$

and suppose $|f(x)| \leq M$ for $x \in E_\rho$. Then $|a_i| \leq 2M\rho^{-i}$ for $i \geq 1$.

Proof. We briefly summarize the proof here, but refer the reader to [Tre19] for more details. Note that for $z \in \mathbb{C}$ with $|z| = 1$, we can identify z in a two-to-one fashion with a point $x \in [-1,1]$ using the formula $x(z) = \frac{1}{2}(z+z^{-1})$, such that $x(z^{-1}) = x(z)$. We further define $F(z) := f\left(\frac{1}{2}(z+z^{-1})\right) = f(x(z))$. Under this transformation, the formula (8) reads

$$a_k = \frac{1}{\pi\iota}\int_{|z|=1} z^{-(i+1)}F(z)\mathrm{d}z = \frac{1}{\pi\iota}\int_{|z|=\rho} z^{-(i+1)}F(z)\mathrm{d}z.$$

To see the latter equality, f is analytic over the interior E_ρ, and $x(z)$ is analytic and sends the annulus $\rho^{-1} \leq |z| \leq \rho$ to E_ρ. Therefore, Cauchy's integral theorem states that we can expand the contour integral to the annulus and not affect the value. Finally, the conclusion follows from

$$|a_k| \leq \frac{1}{\pi}\int_{|z|=\rho} \left|z^{-(i+1)}F(z)\right| \mathrm{d}z \leq \frac{2\pi\rho}{\pi} \cdot \rho^{-(i+1)} \cdot M \leq 2M\rho^{-i}.$$

□

Combining Theorem 3 with Lemma 3 gives another powerful meta-strategy for polynomial approximation. If we can identify an analytic continuation of f beyond $[-1,1]$ (i.e., to a region which dodges poles of f), we immediately get a rate of decay on its Chebyshev coefficients. As an example, consider Problem 1 when $f(x) = x^{-1}$ and $S = [1,\kappa]$, a standard setting for the application described in Section 1.1. We first shift the range to $[-1,1]$, defining

$$g(y) := f(x(y)) = \frac{1}{x(y)}, \text{ where } x(y) := \frac{\kappa-1}{2}y + \frac{\kappa+1}{2}.$$

The pole of f is $x^\star = 0$, so the pole of g is $y^\star = -1 - \frac{2}{\kappa-1} = -1 - \Theta(\frac{1}{\kappa})$ for large κ. Our goal is to find the largest Bernstein ellipse E_ρ so that g is bounded on E_ρ, and $y^\star \notin E_\rho$. The major axis of E_ρ has length $\rho + \rho^{-1}$,[5] so letting $\rho = 1+\epsilon$, we have $\rho + \rho^{-1} \approx 2 + \epsilon^2$. To dodge $y^\star$, we hence need to choose $\epsilon = \Theta(\kappa^{-1/2})$, and it is simple to check this gives $M = O(1)$ in Theorem 3. Applying Theorem 3 then shows $|a_i|$ decays at the rate $\exp(-i\cdot\kappa^{-1/2})$, so degree-$\approx \sqrt{\kappa}$ polynomials sharply approximate the inverse function over $[1,\kappa]$. This matches our intuition from the conjugate gradient method, which is known to converge at a rate $\propto \exp(-i \cdot \kappa^{-1/2})$ in i iterations.

We note that the same calculation applies to any function on $[1,\kappa]$ with a pole at 0, e.g., $f(x) = \sqrt{x}$.

5 Lanczos in exact arithmetic

We sketch a proof of Theorem 1 under exact arithmetic (ignoring issues of finite bit precision, and with $\epsilon = 0$). We begin by describing the algorithm and proving a correctness guarantee, and then discuss the runtime. The Lanczos method run for k iterations produces $\mathbf{Q} \in \mathbb{R}^{d\times k}$ such that $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}_k$ and $\mathbf{Q}$ spans the order-k Krylov subspace $\mathcal{K}_{k-1}$ (defined in (2), with $\mathbf{K} \leftarrow \mathbf{A}$). It then approximates $f(\mathbf{A})\mathbf{x} \approx \mathbf{Q}f(\mathbf{T})\mathbf{Q}^\top\mathbf{x}$, for $\mathbf{T} := \mathbf{Q}^\top\mathbf{A}\mathbf{Q}$. The key observation is:

$$\mathbf{A}^i\mathbf{x} = \mathbf{Q}\mathbf{Q}^\top\mathbf{A}^i\mathbf{x} = \mathbf{Q}\mathbf{Q}^\top\mathbf{A}^i\mathbf{Q}\mathbf{Q}^\top\mathbf{x} = \mathbf{Q}\mathbf{T}^i\mathbf{Q}^\top\mathbf{x} \text{ for } i < k. \tag{9}$$

[5] One way to see this is that E_ρ is the locus of points $\frac{1}{2}(z+z^{-1})$ where $|z| = \rho$, and the major axis corresponds to the largest magnitude on the locus. A calculation shows this occurs when $z = \pm\rho$, and so the axis length is $\rho + \rho^{-1}$.

The first and second equalities follow because $\mathbf{Q}\mathbf{Q}^\top$ is the orthogonal projection to a subspace containing $\mathbf{A}^i\mathbf{x}$ and $\mathbf{x}$. The last equality in (9) holds by

$$\mathbf{Q}\mathbf{T}^i\mathbf{Q}^\top\mathbf{x} = \mathbf{Q}\mathbf{Q}^\top\mathbf{A}\left(\mathbf{Q}\mathbf{T}^{i-1}\mathbf{Q}^\top\mathbf{x}\right) = \mathbf{Q}\mathbf{Q}^\top\mathbf{A}\left(\mathbf{Q}\mathbf{Q}^\top\mathbf{A}^{i-1}\mathbf{Q}\mathbf{Q}^\top\mathbf{x}\right) = \mathbf{Q}\mathbf{Q}^\top\mathbf{A}^i\mathbf{Q}\mathbf{Q}^\top\mathbf{x},$$

where the second equality inducted on i, and the last equality is because $\mathbf{A}^{i-1}\mathbf{Q}\mathbf{Q}^\top\mathbf{x} = \mathbf{A}^{i-1}\mathbf{x} \in \mathcal{K}_{k-1}$. Now, linearly combining (9) implies that for any polynomial p of degree $\leq k-1$, we have

$$p(\mathbf{A})\mathbf{x} = \mathbf{Q}p(\mathbf{T})\mathbf{Q}^\top\mathbf{x}.$$

By applying the triangle inequality and the definition of

$$\delta_k := \min_{p\in\mathcal{P}_{k-1}}\left(\max_{x\in[\lambda_{\min}(\mathbf{A}),\lambda_{\max}(\mathbf{A})]}|p(x)-f(x)|\right),$$

we have for the p attaining the minimum above,

$$\begin{aligned}\left\|f(\mathbf{A})\mathbf{x}-\mathbf{Q}f(\mathbf{T})\mathbf{Q}^\top\mathbf{x}\right\|_2 &\leq \left(\|f(\mathbf{A})-p(\mathbf{A})\|_{\mathrm{op}} + \left\|\mathbf{Q}f(\mathbf{T})\mathbf{Q}^\top - \mathbf{Q}p(\mathbf{T})\mathbf{Q}^\top\right\|_{\mathrm{op}}\right)\|\mathbf{x}\|_2 \\ &= \left(\|f(\mathbf{A})-p(\mathbf{A})\|_{\mathrm{op}} + \|f(\mathbf{T})-p(\mathbf{T})\|_{\mathrm{op}}\right)\|\mathbf{x}\|_2 \leq 2\delta_k\|\mathbf{x}\|_2.\end{aligned}$$

The equality above used orthonormality of $\mathbf{Q}$, and the last inequality used the Cauchy interlacing theorem, which states that $[\lambda_{\min}(\mathbf{T}),\lambda_{\max}(\mathbf{T})] \subseteq [\lambda_{\min}(\mathbf{A}),\lambda_{\max}(\mathbf{A})]$. Note that in exact arithmetic, the $O(k)$ loss in the approximation quality of Theorem 1 is improved to a factor of 2.

We now discuss the runtime of producing the approximation $\mathbf{Q}f(\mathbf{T})\mathbf{Q}^\top\mathbf{x}$ in the Lanczos method. The most important property of $\mathbf{Q}$ from a computational perspective is that it is recursively computed such that $\mathbf{T} := \mathbf{Q}^\top\mathbf{A}\mathbf{Q}$ is tridiagonal, i.e., $\mathbf{T}$ is zero everywhere more than one entry away from the main diagonal. Because $\mathbf{T}$ is tridiagonal, $f(\mathbf{T})$ can be computed in $O(k^2)$ time.

The algorithm computes the $\{\mathbf{q}_j\}_{j\in[k]}$ by iteratively computing $\mathbf{q}_{j+1} \leftarrow \mathbf{A}\mathbf{q}_j$, orthogonalizing it against $\mathbf{q}_j$ and $\mathbf{q}_{j-1}$, and then normalizing it to unit length, which takes $O(\mathcal{T}_{\mathrm{mv}}(\mathbf{A})+d)$ time. By construction, one can verify that $\mathbf{T}$ is tridiagonal, so we need to show $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}_k$.

In other words, our goal is to show that $\mathbf{A}\mathbf{q}_j$ is already orthogonal to $\mathbf{q}_i$ for all $i \in [j-2]$ (so no further orthogonalization is needed). We instead show that $\mathbf{A}\mathbf{q}_i \perp \mathbf{q}_j$, which is equivalent to $\mathbf{A}\mathbf{q}_j \perp \mathbf{q}_i$. Assume inductively that the $\{\mathbf{q}_i\}_{i\in[j]}$ are orthogonal and span $\mathcal{K}_{j-1}$. Because $\mathbf{A}\mathbf{q}_i$ is in the order-$(i+1)$ Krylov subspace $\mathcal{K}_i$, it is in $\mathrm{Span}(\{\mathbf{q}_i\}_{i\in[j-1]})$, and hence $\mathbf{A}\mathbf{q}_i \perp \mathbf{q}_j$ as claimed.

Source material

Portions of this lecture are based on reference material in [SV14, Tre19], as well as the author's own experience working in the field.

References

[AA22] Amol Aggarwal and Josh Alman. Optimal-degree polynomial approximations for exponentials and gaussian kernel density estimation. In *37th Computational Complexity Conference, CCC 2022*, volume 234 of *LIPIcs*, pages 22:1–22:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[FMMS16] Roy Frostig, Cameron Musco, Christopher Musco, and Aaron Sidford. Principal component projection without principal component analysis. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2349–2357. JMLR.org, 2016.

[KV17] Weihao Kong and Gregory Valiant. Spectrum estimation from samples. *Annals of Statistics*, 45(5):2218–2247, 2017.

[MM15] Cameron Musco and Christopher Musco. Randomized block krylov methods for stronger and faster approximate singular value decomposition. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, pages 1396–1404, 2015.

[MMS18] Cameron Musco, Christopher Musco, and Aaron Sidford. Stability of the lanczos method for matrix function approximation. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018*, pages 1605–1624. SIAM, 2018.

[SV14] Sushant Sachdeva and Nisheeth K. Vishnoi. Faster algorithms via approximation theory. *Foundations and Trends in Theoretical Computer Science*, 9(2):125–210, 2014.

[TKV17] Kevin Tian, Weihao Kong, and Gregory Valiant. Learning populations of parameters. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5778–5787, 2017.

[Tre19] Lloyd N. Trefethen. *Approximation theory and approximation practice, extended edition.* Society for Industrial and Applied Mathematics, Philadelphia, PA, 2019. Extended edition [of 3012510].

[TT24] Ewin Tang and Kevin Tian. A CS guide to the quantum singular value transformation. In *2024 Symposium on Simplicity in Algorithms, SOSA 2024*. SIAM, 2024.

[vP15] Tobias von Petersdorff. Amsc/cmsc 666 numerical analysis notes. https://www.math.umd.edu/ petersd/666/amsc666notes02.pdf, 2015. Accessed: 12/10/2023.