Math 6220: Applied Functional Analysis

Derek Lim

Spring 2020

Instructor: Alex Townsend

Course Description: This applied functional analysis class emphasizes the modern uses of approximation theory, reproducing kernel Hilbert spaces, and rational functions. We will cover a selection of theorems in functional analysis and how they are having real-life consequences for modern computations.

Textbooks: Various references

- *Approximation theory and approximation practice* by Nick Trefethen
- *Introduction to the theory of reproducing kernel Hilbert spaces* by Vern Paulsen and Mrinal Raghupathi
- *An Excursion into the Theory of Hankel Operators* by Vladimir Peller
- *A representer theorem for deep neural networks* by Michael Unser

Lecture 1: Introduction (1/21/19)

Topics that we will cover include:

- Peetre's theorem — why differential equations are everywhere.
- Mairhuber-Curtis — why 1D is special in data-fitting.
- Representer theorem — theoretical justification for the effectiveness of ReLU as an activation function in neural networks.
- AAK theory — how eigenvalues and singular values are deeply connected to rational functions.

Goals of the class include putting out notes on applied functional analysis that is readable to a larger audience.

We will start with 1D approximation theory, which is like the nonperiodic analogue to Fourier analysis. Today we will study Chebyshev points and interpolants.

Let $n \geq 0$, and let $P_n =$ the set of polynomials of degree at most n. Say we have $n+1$ distinct points $x_0, \ldots, x_n$ in $[-1, 1]$, and sample values $f_0, \ldots, f_n \in \mathbb{C}$. We want a $p \in P_n$ such that $p(x_j) = f_j$ for $0 \leq j \leq n$.

Theorem 1. *There exists a unique $p \in P_n$ such that $p(x_j) = f_j$, $0 \le j \le n$.*

Proof. An interpolating polynomial is given by a solution to the Vandermonde system. The Vandermonde matrix is invertible for unique nodes x_j, so there exists a unique solution. □

Now, instead of being given nodes and sample values, say we are given a continuous function $f : [-1, 1] \to \mathbb{R}$. We wish to replace it by a polynomial $p \approx f$. Polynomials are signficantly easier to work with than a general continuous function. For instance, differentiation, integration, evaluation, and storage are simple. We have two choices to make in this approximation:

- Pick sample nodes $x_0, \ldots, x_n$
- A basis to represent p

The monomial power basis of course is simple and is nice to work with theoretically, but it is terrible for numerical usage. x^{100} is very close to x^{102} in $[-1, 1]$, even though they are linearly independent as functions. Also, equally spaced nodes are a poor choice numerically.

Instead, we use **Chebyshev points** as nodes, which are of the form $x_j = \cos\left(\frac{j\pi}{n}\right)$. In Fourier analysis, we would take equally spaced nodes for a function defined on the circle. Here, we project these down to 1D by taking the cos of these points. Thus, the nodes are more clustered towards the endpoints -1 and 1. Later in the course we will give justification for why these are a good choice of nodes. We define the Chebyshev interpolant as the polynomial p that matches f on these nodes: $p(x_j) = f(x_j)$, $0 \le j \le n$.

For Fourier/Laurent theory, we have:

- $z \in$ unit circle
- For preciseness, we assume we have symmetry: $f(z)$ with $f(z) = f(z^{-1})$
- Representation in a basis

$$\begin{aligned} f(\theta) &= \frac{1}{2} \sum_{k=0}^{\infty} a_k (e^{ik\theta} + e^{-ik\theta}) \\ &= \sum_{k=0}^{\infty} a_k \cos(k\theta) \end{aligned}$$

For the Chebyshev theory, we have:

- $x \in [-1, 1]$
- $f(x) : [-1, 1] \to \mathbb{R}$
- Representation in a basis

$$f(x) = \sum_{k=0}^{\infty} c_k \cos(k \cos^{-1}(x))$$

The $T_k(x) = \cos(k \cos^{-1} x)$ are the **Chebyshev polynomials**.

For the Chebyshev nodes x_j and the Chebyshev basis $T_k(x)$, the linear system to solve for the interpolant is

$$\begin{bmatrix} T_0(x_0) & T_1(x_0) & \dots & T_n(x_0) \\ \vdots & \vdots & & \vdots \\ T_0(x_n) & T_1(x_n) & \dots & T_n(x_n) \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(x_0) \\ \vdots \\ f(x_n) \end{bmatrix}$$

This is the discrete cosine transform matrix, as $T_k(x_j) = \cos(k \cos^{-1}(\cos(j\pi/n))) = \cos(jk\pi/n)$. Solves with this matrix can be done quickly, in $\mathcal{O}(n \log n)$.

Demo with Chebyshev approximation to $\sin(x) + \sin(x^2)$.

Theorem 2. *Let $f : [-1, 1] \to \mathbb{R}$, be Lipschitz continuous. Then f has a unique representation as*

$$f(x) = \sum_{k=0}^{\infty} a_k T_k(x)$$

and the series is absolutely and uniformly convergent. Moreover,

$$a_k = \frac{1}{2\pi} \int_{-1}^{1} \frac{f(x) T_k(x)}{\sqrt{1 - x^2}}\, dx, \quad k \geq 1$$
$$a_0 = \frac{1}{\pi} \int_{-1}^{1} \frac{f(x) T_0(x)}{\sqrt{1 - x^2}}\, dx$$

Lecture 2: Chebyshev Interpolation and Projection (1/23)

Given last lecture, we have two ways of computing a polynomial approximant to f:

- Interpolation
$$p_n(x) = \sum_{k=0}^{n} c_k T_k(x)$$
where c_k are chosen so that $p_n(x_j) = f(x_j)$, at the Chebyshev points x_j.
- Projection (truncation)
$$f_n(x) = \sum_{k=0}^{n} a_k T_k(x)$$
where a_k are given as in theorem 2.

The interpolation method is extremely easy and efficient to compute with FFT methods. The projection method is difficult to compute (have to evaluate integrals where the integrand has a singularity and are highly oscilatory), but easy to analyze. The projection is often better in error than the interpolant at the same n. However, it is much faster to compute the interpolant, so it would be used at higher n.

The following theorem describes aliasing in Chebyshev interpolation.

Theorem 3. *Let $n \geq 1$ and Chebyshev points $x_j = \cos(j\pi/n)$ for $0 \leq j \leq n$. For any $0 \leq m \leq n$, the following Chebyshev polynomials are the same on x_j: T_m, T_{2n-m}, T_{2n+m}, T_{4n-m}, T_{4n+m}, T_{6n-m}, ...*

Proof. The proof is simple; we prove a special case.

$$\begin{aligned} T_{2n+m}(x) &= \Re(e^{i(2n+m)j\pi/n}) \\ &= \Re(e^{imj\pi/n}) \\ &= T_m(x_j) \end{aligned}$$

□

Theorem 4. *If f is Lipschitz continuous, then*

$$\begin{aligned} c_0 &= a_0 + a_{2n} + a_{4n} + \dots \\ c_n &= a_n + a_{3n} + a_{5n} + \dots \\ c_m &= a_m + (a_{2n-m} + a_{4n-m} + \dots) + (a_{2n+m} + a_{4n+m} + \dots) \end{aligned}$$

Proof. The proof is by grouping like terms in the Chebyshev series.

$$\begin{aligned} f(x_j) &= \sum_{k=0}^{\infty} a_k T_k(x_j) \\ &= (a_0 + a_{2n} + a_{4n} \dots)T_0(x_j) + [a_1 + (a_{2n-1} + a_{4n-1} + \dots) + (a_{2n+1} + \dots)]T_1(x_j) + \dots + c_n T_n(x_j) \end{aligned}$$

The polynomial interpolant is just this last term as a function of x, as it is of degree at most n and matches f at the nodes. □

Corollary 1. For $x \in [-1, 1]$,

$$f(x) - f_n(x) = \sum_{k=n+1}^{\infty} a_k T_k(x)$$

$$\left|f(x) - f_n(x)\right| \leq \sum_{k=n+1}^{\infty} |a_k|$$

$$f(x) - p_n(x) = \sum_{k=n+1}^{\infty} a_k (T_k(x) - T_s(x))$$

where $s = \left|(k + n - 1) \bmod 2n - (n-1)\right|$

This corollary means that p_n is about 2 times worse than f_n.

Example 0.1. For the Chebyshev series of $x \mapsto |x|$, the coefficients can be solved analytically:

$$a_0 = \frac{2}{\pi}, \quad a_2 = \frac{4}{3\pi}, \quad a_4 = \frac{-4}{15\pi}, \quad \dots$$

The Chebyshev series of $x \mapsto e^x$ are Bessel functions $a_0 = I_0(1), \quad a_k = 2I_k(1)$. Analyzing these shows that one needs a polynomial of degree 14 to represent e^x up to machine precision. This causes some issues. For instance, differentiating this polynomial 15 times gives the zero function, even though it should give e^x. However, numerical differentiation is already very numerically difficult.

Theorem 5. *Let $f \in [-1,1] \to \mathbb{R}$ be continuous.*

$$\|f(x) - p_n(x)\|_\infty \leq \left[2 + \frac{2}{\pi}\log(n+1)\right] \|f(x) - p_n^{\text{best}}(x)\|_\infty$$

where p_n^{best} minimizes the sup norm over $[-1,1]$.

The difference between the errors is small, for instance at $n = 100$, it the coefficient is 4.9381, and at $n = 100000$, it is 10.7952.

We now explain how to evaluate the Chebyshev interpolant quickly by a method of Clenshaw's. The recurrence is

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x)$$

We write recurrence relations in a linear system.

$$\underbrace{\begin{bmatrix} 1 & & & & \\ -x & 1 & & & \\ & \ddots & & & \\ & & 1 & -2x & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} T_0(x) \\ T_1(x) \\ \vdots \\ T_n(x) \end{bmatrix}}_{\vec{T}_n(x)} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

To evaluate at x_*, we have

$$\begin{aligned} p_n(x_*) &= \sum_{k=0}^{n} c_k T_k(x_*) \\ &= \vec{T}_n(x_*)^T \vec{c} \\ &= e_1^T (L^{-T} \vec{c}) \end{aligned}$$

so that all is required is a backsolve of this sparse triangular matrix.

Lecture 3: Weierstrass, Convergence of Coefficients (1/28)

It is simple to compute with Chebyshev approximations:

- Integration

$$\begin{aligned} \int_{-1}^{1} f(x)\,dx &\approx \int_{-1}^{1} p_n(x)\,dx \\ &= \sum_{k=0}^{n} c_k \int_{-1}^{1} T_k(x)\,dx \\ &= 2c_0 + \sum_{k=0}^{\lfloor n/2 \rfloor} c_{2k} \frac{2}{1-(2k)^2} \end{aligned}$$

- Root finding can be done by finding the eigenvalues of a certain matrix that has characteristic polynomial equal to the interpolant. Standard eigensolvers can be used on these sparse matrices. There is also a trick in which the interpolation is done by piecewise polynomials, so the dimension of each subproblem is smaller and suffers less from the $\mathcal{O}(n^3)$ scaling of eigensolvers.

Theorem 6. *The interpolant p_n is the characteristic polynomial $p_n(x) = \det(xI - C_n)$ where*

$$C_n = \begin{bmatrix} 0 & 1/2 & & \\ 1/2 & 0 & 1/2 & \\ & \ddots & \ddots & 1/2 \\ & & 1/2 & 0 \end{bmatrix} - \frac{1}{2c_n} \begin{bmatrix} & & & \\ & & & \\ & & & \\ c_0 & c_1 & \dots & c_{n-1} \end{bmatrix}$$

Weierstrass Approximation Theorem

Theorem 7 (Weierstrass)**.** *Given $f : [-1, 1] \to \mathbb{R}$ continuous and $\epsilon > 0$, there exists a polynomial p such that*

$$\|f - p\|_\infty < \epsilon$$

Theorem 8 (Faber)**.** *No interpolant has this property.*

Proof of Weierstrass. There are several proofs of the Weierstrass theorem. One is by Bernstein. The original proof has nice physical interpretations (viewing f as heat on a rod and running the heat equation for a small moment), and takes the following steps:

1. Extend f to $F : \mathbb{R} \to \mathbb{R}$ continuous with compact support.
2. Run the heat equation with F as the initial condition for a short time (i.e. convolving F with a narrow Gaussian kernel). This gives us an analytic function G.
3. Take p to be a suitable truncation of the Taylor series of G.

□

Note that the method of the proof is terrible to compute with. The schemes from other proofs also have lower convergence rates than those of Chebyshev approximations. This is because these schemes have to be general and work for all types of continuous functions.

Convergence of Chebyshev coefficients

A Jackson-type statement, which rougly states "The smoother the f, the faster the Chebyshev series coefficients converge":

- If f has k derivatives (plus a slightly stronger condition), then $|a_n| = \mathcal{O}(n^{-k})$.
- If f is analytic, $|a_n| = \mathcal{O}(\rho^{-n})$ for some $\rho > 1$.

Theorem 9. *Let $\nu \geq 0$ such that $f, f', \dots, f^{(\nu-1)}$ are continuous and $f^{(\nu)}$ has bounded variation. Then*

$$|a_k| \leq \frac{2V}{\pi(k-\nu)^{\nu+1}} \qquad k \geq \nu + 1$$

where V is the total variation of $f^{(\nu)}$.

If $\nu \geq 1$,

$$\|f - f_n\|_\infty \leq \frac{2V}{\pi\nu(n-\nu)^\nu}, \qquad \|f - p_n\|_\infty \leq \frac{4V}{\pi\nu(n-\nu)^\nu}$$

The assumption of bounded variation is necessary to get an extra power of n for the convergence.

Definition 0.1. The variation of $f : [-1,1] \to \mathbb{R}$ is

$$V(f) = \sup_{-1 \le y_1 \le \dots \le y_m \le 1} \sum_{i=1}^{m-1} |f(y_{i+1}) - f(y_i)|$$

If f has a continuous derivative, this turns out to be $V(f) = \int_{-1}^{1} |f'(x)| \, dx$.

Convergence in the Analytic Case

If f is analytic on $[-1,1]$, then it has a convergent Taylor series in a neighborhood about x for any $x \in [-1,1]$. By compactness we can take a finite subcover of $[-1,1]$. Then there is a Berstein ellipse with focal points at -1 and 1 which is contained within the subcover.

Theorem 10. *If $f : [-1,1] \to \mathbb{R}$ is analytic and analytically continuable to an open Berstein ellipse of size $\rho > 0$, then*

$$|a_k| < 2M\rho^{-k} \qquad k \ge 0$$

where $M = \sup_{z \in E_\rho} |f(z)|$, for E_ρ the Berstein ellipse.

Also, we have

$$\|f - f_n\|_\infty \le \frac{2M\rho^{-n}}{\rho - 1}, \qquad \|f - p_n\|_\infty \le \frac{4M\rho^{-n}}{\rho - 1}$$

Lecture 4: Approximation in Different Norms (1/30)

There are multiple ways to replace f by a proxy $p \in P_n$.

- Best L_∞ fit. $\|f\|_\infty = \max_{x \in [-1,1]} |f(x)|$. Then $p_n^{L^\infty} = \operatorname{argmin}_{q \in P_n} \|f - q\|_\infty$.
- Best L_2 fit. $\|f\|_2^2 = \int_{-1}^{1} |f(x)|^2 \, dx$. Then $p_n^{L^2} = \operatorname{argmin}_{q \in P_n} \|f - q\|_2$.
- Best L_1 fit. $\|f\|_1 = \int_{-1}^{1} |f(x)| \, dx$. Then $p_n^{L^1} = \operatorname{argmin}_{q \in P_n} \|f - q\|_1$.
- Chebyshev interpolant. $p_n^{(\text{cheb})}(x_k) = f(x_k), \ 0 \le k \le n, \ x_k = \cos(k\pi/n)$

Definition 0.2. We say $f - q$ **equioscillates** $n+2$ times if there are points $\xi_0, \dots, \xi_{n+1} \in [-1,1]$ such that $f(\xi_i) - q(\xi_i) = -(f(\xi_{i-1}) - q(\xi_{i-1})), \quad 1 \le i \le n+1$.

Theorem 11 (Equioscillation)**.** *$p_n^{L^\infty}$ is the best approximation in the L_∞ norm if and only if there are $n+2$ equioscillation points $\xi_0, \dots, \xi_{n+1} \in [-1,1]$ such that $(f - p_n^{L^\infty})(\xi_k) = \pm(-1)^k \left\| f - p_n^{L^\infty} \right\|_\infty$*

Proof. Suppose p is a polynomial of degree at most n such that $(f-p)(\xi_k) = \pm(-1)^k \|f - p\|_\infty$. Suppose there is a $q \in P_n$ such that $\|f - q\|_\infty < \|f - p\|_\infty$. Since $f - p$ equioscillates $n+2$ times, then $f - q$ must match $f - p$ at $n+1$ points. Then $q = p$ since they are both degree at most n, giving a contradiction.

The other direction is a bit less intuitive (see ATAP). □

While this is a really elegant theorem, note that this means $p_n^{L_\infty}$ achieves its largest error across the whole range of the equioscillation points. For instance, the Chebyshev interpolant of $x \mapsto |x|$ stays within a smaller error range for much of the interval $[-1,1]$, while the best L_∞ interpolant hits the highest L_∞ errors at points spread across the whole interval.

Theorem 12. *$p_n^{L_2}$ is the best L_2 approximant if and only if $\langle f - p_n^{L_2}, q\rangle_{L_2} = 0$ for any $q \in P_n$.*

Proof. If $\langle f - p_n^{L_2}, q\rangle = 0$ for all $q \in P_n$, then for any $\tilde{p}_n \in P_n$, set $s = p_n^{L_2} - \tilde{p}_n$. Then we have

$$\begin{aligned}
\|f - \tilde{p}_n\|_2^2 &= \left\|f - p_n^{L_2} + s\right\|_2^2 \\
&= \left\|f - p_n^{L_2}\right\|_2^2 + \|s\|_2^2 && s \in P_n \\
&\geq \left\|f - p_n^{L_2}\right\|_2^2
\end{aligned}$$

with equality if and only if $s = 0$.

Conversely, if $p_n^{L_2}$ is best, but $\langle f - p_n^{L_2}, q\rangle \neq 0$ for some $q \in P_n$, then for $\lambda = -\frac{\langle f - p_n^{L_2}, q\rangle}{\|q\|_2}$, consider the polynomial $p_n^{L_2} - \lambda q \in P_n$. This in fact is a better approximation to f in the L_2 sense. □

One would particularly like to use L_2 norm for approximation for noisy data; Gaussian noise is orthogonally invariant.

To compute $p_n^{L_2}$, we want to construct an orthgonal (in the L_2 sense) basis for $\mathbb{R}_n[x]$. An orthogonal basis for $\mathbb{R}_n[x]$ is the Legendre polynomials $P_0, \ldots, P_n$. They satisfy

$$\begin{aligned}
\langle P_j, P_k\rangle &= \int_{-1}^{1} P_j(x) P_k(x)\, dx \\
&= \begin{cases} 0 & j \neq k \\ \frac{2}{2j+1} & j = k \end{cases}
\end{aligned}$$

Then we have $f(x) = \sum_{k=0}^{\infty} b_k P_k(x)$ where $b_k = \frac{\langle f, P_k\rangle}{\langle P_k, P_k\rangle}$. This equality in the series (for integrable f) means equality in L_2. Just as in the Chebyshev case, if f is Lipschitz, then the series is absolutely and uniformly convergent. The best L_2 approximation is then $p_n^{L_2}(x) = \sum_{k=0}^{n} b_k P_k(x)$. Thus, this can be computed with just some integrals, as opposed to in the L_∞ case where a massive optimization problem must be solved.

However, integrals are still expensive, so in practice, we replace this projection with an interpolant. Here we use Legendre interpolants. This is in analogy with the Chebyshev case, in which we can more readily analyze the projection, but compute at the interpolant. The Legendre interpolants are those at the roots of the $n+1$th Legendre polynomial:

$$p_n^{\text{leg}}(x_i^{\text{leg}}) = f(x_i^{\text{leg}}), \quad 0 \leq i \leq n$$

The x_i^{leg} are selected as the nodes from the Gauss-Legendre quadrature, meaning $P_{n+1}(x_i^{\text{leg}}) = 0$, $0 \leq i \leq n$.

Lecture 5: Quadrature (2/4)

Recall from last time that the best approximation in the L_∞ norm to f satisfies that $f - p_n^{L_\infty}$ has $n+2$ equioscillation points, so there are at least $n+1$ zeros, and thus $p_n^{L_\infty}$ is an interpolant to f. However, the interpolation nodes depend nonlinearly on f. The best Chebyshev interpolant to $f+g$ is the sum of the Chebyshev interpolants of f and g.

Interpolation

Given $f \in C[-1,1]$, we want to compute $I = \int_{-1}^{1} f(x)\,dx$. We have distinct sample nodes $x_0, x_1, \ldots, x_n$ and weights $w_0, w_1, \ldots, x_n$. The integral is approximated by $I \approx I_n = \sum_{k=0}^{n} w_k f(x_k)$. Usually, these weights come from some polynomial interpolation scheme.

An interpolative quadrature rule approximates

$$\int_{-1}^{1} f(x)\,dx \approx \int_{-1}^{1} p_n(x)\,dx = \sum_{k=0}^{n} w_k p(x_k) = \sum_{k=0}^{n} w_k f(x_k)$$

where p_n is a polynomial interpolant to f at points $x_0, \ldots, x_n$, which justifies the last equality. The middle equality needs to be enforced by a choice of w_k.

Three popular quadrature rules:

- Newton-Cotes: x_j are equally spaced.

  Equally spaced nodes are very good for periodic functions. In particular, the trapezoidal rule can work geometrically well. However, $|I - I_n|$ can diverge as $n \to \infty$ (Runge was the first to show this).

- Clenshaw-Curtis: x_j are the Chebyshev points.

  This can be computed quickly with FFT

- Gauss quadrature: x_j are the Legendre points.

Theorem 13. *For any interpolative quadrature rule, $I = I_n$ is exact if $f \in \mathbb{R}_n[x]$.*

For Gauss quadrature, $I = I_n$ is $f \in \mathbb{R}_{2n+1}[x]$.

Proof. The first part is simply due to the fact that f equals its degree n interpolant.

For Gauss quadrature, let $f \in P_{2n+1}$. Then $f(x) = q_n(x)P_{n+1}(x) + r_n(x)$ where $q_n, r_n \in \mathbb{R}_n[x]$, and P_{n+1} is the Legendre polynomial. Integrating gives

$$\begin{aligned}
\int_{-1}^{1} f(x)\,dx &= \int_{-1}^{1} q_n(x)P_{n+1}(x)\,dx + \int_{-1}^{1} r(x);dx \\
&= \int_{-1}^{1} r(x)\,dx && \text{orthogonality} \\
&= \sum_{k=0}^{n} w_k r(x_k) \\
&= \sum_{k=0}^{n} w_k f(x_k)
\end{aligned}$$

where the last equality is since $f(x_k) = q_n(x_k)P_{n+1}(x_k) + r_n(x_k) = r_n(x_k)$ by choice of Legendre points as x_k. $\square$

Gauss quadrature is the best quadrature rule in this narrow sense. However, for different types of functions (such as analytic functions), there are better quadrature rules.

Implementation of Clenshaw-Curtis

$$\int_{-1}^{1} T_k(x)\,dx = \begin{cases} 0 & k \text{ odd} \\ \frac{2}{1-k^2} & k \text{ even} \end{cases}$$

Thus, we can easily compute

$$\begin{aligned} \int_{-1}^{1} f(x)\,dx &\approx \int_{-1}^{1} \sum_{k=0}^{n} c_k T_k(x)\,dx \\ &= \sum_{k=0}^{n} c_k \int_{-1}^{1} T_k(x)\,dx \\ &= \sum_{k=0}^{\lfloor n/2 \rfloor} c_{2k} \frac{2}{1-(2k)^2} \end{aligned}$$

Note that after computing the interpolant, we do not need to sample f again to compute the integral.

Implementation of Gauss quadrature

This method was introduced by Golub and Welsch in 1969. The idea is to find the roots of $P_{n+1}(x)$ by an eigenvalue problem. We do this by using the recurrence relation

$$P_0(x) = 1, \quad P_1(x) = x, \quad (n+1)P_{n+1}(x) = (2n+1)P_n(x) - nP_{n-1}(x) \quad n \geq 1$$

Then we set up the eigenvalue problem

$$\begin{bmatrix} 0 & 1 & & & \\ \frac{1}{3} & 0 & \frac{2}{3} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{n-1}{2n-1} & 0 & \frac{n}{2n-1} \\ & & & \frac{n}{2n+1} & 0 \end{bmatrix} \begin{bmatrix} P_0(\lambda) \\ P_1(\lambda) \\ \vdots \\ P_n(\lambda) \end{bmatrix} = \lambda \begin{bmatrix} P_0(\lambda) \\ P_1(\lambda) \\ \vdots \\ P_n(\lambda) \end{bmatrix}$$

at $n = 1$ for instance, we have $\frac{2}{3}P_2(x) + \frac{1}{3}P_0(x) = xP_1(x)$. Then on the last row, we want λ to be a root of P_{n+1}, so we have

$$\begin{aligned} 0 &= (2n+1)\lambda P_n(\lambda) - nP_{n-1}(\lambda) \\ \frac{n}{2n+1}P_{n-1}(\lambda) &= \lambda P_n(\lambda) \end{aligned}$$

Then this eigenvalue problem can be solved in $O(n^2)$ by symmetrizing the tridiagonal system. However, there are formulas for computing the Legendre points up to machine precision that are much quicker.

Accuracy of quadrature rules

We analyze the error $E_n(f) = I(f) - I_n(f)$ of Gauss quadrature. If f is Lipschitz continuous, then we can write $f(x) = \sum_{k=0}^{\infty} a_k T_k(x)$. Then we have

$$\begin{aligned}
|E_n(f)| &= \left| \sum_{k=0}^{\infty} a_k E_n(T_k) \right| \\
&= \left| \sum_{k=2n+2} a_k E_n(T_k) \right| && \text{integrates } \mathbb{R}_{2n+1}[x] \text{ exactly} \\
&= \left| \sum_{\substack{k=2n+2 \\ k \text{ even}}} a_k E_n(T_k) \right| \\
&\leq \sum_{\substack{k=2n+2 \\ k \text{ even}}} |a_k||E_n(T_k)| \\
&\leq 4 \sum_{\substack{k=2n+2 \\ k \text{ even}}} |a_k|
\end{aligned}$$

where we use the Gauss quadrature integrates odd Chebyshev polynomials exactly. This can be checked, note that odd Chebyshev polynomials integrate to 0.

Quadrature and rational functions

Claim: Every quadrature rule takes the following scheme:

1. Approximate $f(x)$ by a rational interpolant.
2. Integrate the interpolant exactly.

Note that in the case of interpolative quadrature rules, the interpolants are polynomials.

Say we have a quadrature rule $\int_{-1}^{1} f(x)\,dx \approx \sum_{k=0}^{n} w_k f(x_k)$. Recall that we have the Lagrange form of f. Let $l(x) = \prod_{k=0}^{n}(x - x_k)$. Define Lagrange weights $v_k = \frac{1}{\prod_{i \neq k}(x_k - x_i)}$. Then any polynomial q can be written $q(x) = l(x) \sum_{k=0}^{n} \frac{v_k}{x - x_k} q(x_k)$.

Let $r(x) = \frac{p(x)}{q(x)}$ be an interpolant. Then $r(x_j) = f(x_j)$ means $p(x_j) = f(x_j)q(x_j)$. Note that then we can write $p(x) = l(x) \sum_{k=0}^{n} \frac{v}{x - x_k} f(x_k) q(x_k)$. Defining $\mu_k = v_k q(x_k)$, we have

$$\begin{aligned}
r(x) &= \frac{l(x) \sum_{k=0}^{n} \frac{v_k}{x - x_k} f(x_k) q(x_k)}{l(x) \sum_{k=0}^{n} \frac{v_k}{x - x_k} q(x_k)} \\
&= \frac{l(x) \sum_{k=0}^{n} \frac{\mu_k}{x - x_k} f(x_k)}{l(x) \sum_{k=0}^{n} \frac{\mu_k}{x - x_k}}
\end{aligned}$$

Then the integration is

$$\int_{-1}^{1} r(x)\,dx = \sum_{k=0}^{n} \underbrace{\int_{-1} \frac{\frac{\mu}{x-x_k}}{\sum_{j=0}^{n} \frac{\mu_j}{x-x_j}}\,dx}_{w_k} f(x_j)$$

Example 0.2 (Monte Carlo)**.** Say we have the scheme:

1. Sample $f(x)$ at $x_0,\ldots,x_n$ (random locations)
2. $I = \int_{-1}^{1} f(x)\,dx \approx 2\frac{1}{n+1}\sum_{k=0}^{n} f(x_k)$

This is equivalent to

1. Approximate $f(x)$ by a constant (in the least squares sense)
2. Integrate the constant exactly

For the first step, we want to solve $y = c$, meaning

$$\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} c = \begin{bmatrix} f(x_0) \\ \vdots \\ f(x_n) \end{bmatrix}$$

Since this does not generally have a solution, we take the best least squares fit, meaning $A^TAc = A^Tb$. Then we have that $(n+1)c = \sum_{k=0}^{n} f(x_k)$.

Lecture 6: (2/6)

We have gone over very strong results for interpolant approximation in 1D. However, there are issues with interpolant approximation in higher dimensions.

Theorem 14 (Mairhuber-Curtis)**.** *Given a basis $\{\phi_1,\ldots,\phi_N\}$, a function $f:\Omega\to\mathbb{R}$, and samples $x_1,\ldots,x_N\in\Omega$ distinct, then if Ω contains an interior point, it may not be possible to construct an interpolant of the form*

$$p_N(x) = \sum_{i=1}^{N} c_i\phi_i(x) \quad \text{s.t.} \quad p_N(x_i) = f(x_i)\ 1 \le i \le N$$

Proof. The linear system for making an interpolant is

$$\underbrace{\begin{bmatrix} \phi_1(x_1) & \ldots & \phi_N(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_N) & \ldots & \phi_N(x_N) \end{bmatrix}}_{A} \begin{bmatrix} c_1 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix}$$

so p_N exists if and only if A is invertible. Choose the samples x_j such that x_1 and x_2 are in a ball within Ω. If A has determinant zero, then we are done. Otherwise, take a path from x_1 to x_2 and a nonintersecting path from x_2 to x_1. Moving the points between these paths continuously interchanges two rows of A, thus negating the determinant. Since the determinant is continuous, there is a point where the determinant is zero. □

Note that this proof does not work in 1D, since any two such paths will intersect in an interval. Also, this proof reveals that any choice of samples with two inside a ball contained in Ω is quite close to not having an interpolant through them.

We can either

- Choose $x_1, \dots x_N \in \Omega$ given a basis $\{\phi_1, \dots, \phi_N\}$
- Choose $\{\phi_1, \dots, \phi_N\}$ given a point set $x_1, \dots, x_N \in \Omega$

Kernels

Using the second method, we consider a function $K : \Omega \times \Omega \to \mathbb{R}$. We want to try to interpolate

$$p_N(x) = \sum_{k=1}^{N} c_k K(x, x_k) \qquad p_N(x_k) = f(x_k)\ 1 \le k \le N$$

note then that the basis depends on the samples, so the argument in the proof of Mairhuber-Curtis does not work. In particular, continuously swapping two samples changes the whole matrix. We want the kernel matrix, $A_{ij} = K(x_i, x_j)$ to be invertible.

It is too hard to specify the family of kernels for which A^{-1} exists (we still do not know!). An easier way is to take A to be symmetric positive definite. For this to hold, we need our kernels to be **positive definite**, meaning that

- K is symmetric, i.e. $K(x, y) = K(y, x)$
- For any $x_1, \dots, x_N \in \Omega$, the kernel matrix A is positive definite

K is positive definite if and only if for every $x_1, \dots, x_N \in \Omega$,

$$c^T A c = \sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j K(x_i, x_j) > 0 \quad c \neq 0$$

Example 0.3 (Linear kernel)**.** Let $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be given by $K(x, y) = x^T y$ is a positive definite kernel. This is because for any $x_1, \dots, x_N \in \mathbb{R}^d$, the kernel matrix is $X^T X$ where $X = \begin{bmatrix} x_1 & \dots & x_N \end{bmatrix}$

Example 0.4 (Gaussian)**.** Let $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be given by $K(x, y) = e^{-\frac{\|x-y\|_2^2}{2\sigma^2}}$. This is one of the most useful positive definite kernels.

We must also ask: given a kernel function, what functions can we learn? Given a function $f : \Omega \to \mathbb{R}$, anything of the form $f(x) = \sum_{k=1}^{N} c_k K(x, y_k)$ can obviously be approximated. Thus, we define $H_K = \text{span}\{K(\cdot, x) \text{ for any } x \in \Omega\}$.

Lecture 7: (2/11)

Missed this lecture due to illness.

Definition 0.3. Let $K : \Omega \times \Omega \to \mathbb{R}$ be a positive definite kernel for a Hilbert space H of functions on Ω. We say that K is a **reproducing kernel** if for every $f \in H$ and all $x \in \Omega$,

$$f(x) = \langle f, K(\cdot, x) \rangle$$

Example 0.5. Let $X = \{x_1, \ldots, x_N\}$ be a discrete set, and define

$$l^2(X) = \{f : X \to \mathbb{R} \mid \sum_{x \in X} |f(x)|^2 < \infty\}$$

Then we can define an inner product on $l^2(X)$ by $\langle f, g \rangle = \sum_{x \in X} f(x)g(x)$. We want a kernel K such that for every $f \in l^2(X)$ and $x \in X$,

$$f(x) = \langle f, K(\cdot, x) \rangle = \sum_{y \in X} f(y)K(y, x)$$

Note that $K(y, x) = \delta_{xy}$ satisifies this condition. In this case the kernel matrix is the identity matrix.

Theorem 15 (Moore-Aronszajn)**.** *Let $K : \Omega \times \Omega \to \mathbb{R}$ be a symmetric positive definite kernel. Then there exists a unique Hilbert space of functions on Ω for which K is a reproducing kernel.*

Proof. Let $H_0 = \text{span}\{K(\cdot, x) \mid x \in \Omega\}$. Define the inner product on H_0 by

$$\Big\langle \sum_{i=1}^{m} a_i K(\cdot, x_i), \sum_{j=1}^{n} b_j K(\cdot, y_j) \Big\rangle = \sum_{i=1}^{m} \sum_{j=1}^{n} a_i b_j K(x_i, y_j)$$

The fact that this defines a symmetric inner product is due to K being symmetric and positive definite. Take H to be the completion of H_0 with respect to the topology induced by this inner product.

From here, one may check that K is a reproducing kernel and that H is unique. □

Deriving ReLU as a reproducing kernel

We consider a basic neural net, in which given an input X we have

$$\begin{aligned} L_0(X) &= X \\ L_n(X) &= \sigma_n(w_n L_{n-1}(X)) \qquad\qquad n \geq 1 \\ f(X) &= g(L_N(x)) \end{aligned}$$

The σ_n are activation functions. A common choice of g is the softmax, but g depends on what type of output one would like from their net. Historically, activations have been chosen to be sigmoids, thresholding, or tanh functions. A common choice now is the Rectified Linear Unit — the ReLU — given by $\sigma(x) = \max(x, 0)$. Later a theoretical justification for using ReLU will be presented.

Lecture 8: (2/13)

Missed this lecture due to illness.

Example 0.6 (Examples of Positive Definite Kernels)**.** Here are some examples of positive definite kernels:

- Linear: $K(x,y) = x^T y$
- Polynomial: $K(x,y) = (x^T y + r)^n$, $r \geq 0$
- Gaussian: $K(x,y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$
- Laplacian: $K(x,y) = e^{-\alpha\|x-y\|}$, $\alpha > 0$
- Matérn: $K(x,y) = \|x-y\|_2^{k-\frac{d}{2}} B_{\frac{d}{2}-k}\|x-y\|_2^2$ where k, d are integers and $k > \frac{d}{2}$.
  - B is the Bessel function of the third kind.
  - This is the reproducing kernel for the Sobolev space $W_2^k(\mathbb{R}^d)$.

We now state the Representer Theorem, which allows us to turn certain infinite-dimensional optimization problems into finite-dimensional optimization problems. Say we are given

- Training examples: $x_1, \ldots, x_n \in \Omega$.
- Training values: $y_1, \ldots, y_n \in \mathbb{R}$.
- H an RKHS of functions from ω to $\mathbb{R}$.
- Error function: $E : (\Omega \times \mathbb{R} \times \mathbb{R})^n \to \mathbb{R} \cup \{\infty\}$.
- Strictly monotonically increasing function: $g : [0,\infty) \to \mathbb{R}$.

Theorem 16 (Representer Theorem)**.** *Any minimizer*

$$f^* = \operatorname{argmin}_{f \in H} E\left((x_1, y_1, f(x_1)), \ldots, (x_n, y_n, f(x_n))\right) + g(\|f\|_H)$$

has a representation

$$f^*(x) = \sum_{i=1}^n \alpha_i K(x, x_i) \qquad \alpha_i \in \mathbb{R}, i = 1, \ldots, n$$

Proof. Let $f \in H$. Then we can write $f = \sum_{i=1}^n \alpha_i K(x, x_i) + v$, where v is orthogonal to the space spanned by the $K(\cdot, x_i)$. Thus, we have by the reproducing property that

$$\begin{aligned} f(x_j) &= \left\langle \sum_{i=1}^n \alpha_i K(\cdot, x_i) + v, K(\cdot, x_j) \right\rangle \\ &= \sum_{i=1}^n \alpha_i \langle K(\cdot, x_i), K(\cdot, x_j) \rangle \end{aligned}$$

so the minimization of the summand involving E is independent of v. Now, looking at the regularization term, we have that

$$\begin{aligned} g(\|f\|) &= g\left(\left\|\sum_{i=1}^{n}\alpha_i K(\cdot,x_i)+v\right\|\right) \\ &= g\left(\sqrt{\left\|\sum_{i=1}^{n}\alpha_i K(\cdot,x_i)\right\|^2+\|v\|^2}\right) \\ &\geq g\left(\left\|\sum_{i=1}^{n}\alpha_i K(\cdot,x_i)\right\|\right) \end{aligned}$$

so a minimizing choice of f occurs when $v=0$. □

Lecture 9: Applications of Representer Theorem (2/18)

Example 0.7. Here is an application of the representer theorem to the case of regularized least squares. Suppose we have

$$E=\sum_{i=1}^{n}(y_i-f(x_i))^2 \qquad g\left(\|f\|_H\right)=\lambda\|f\|_H^2,\ \lambda>0$$

Note that the representer theorem gives

$$\begin{aligned} \|f\|_H^2 &= \langle f,f\rangle \\ &= \langle\sum_{i=1}^{n}a_i K(\cdot,x_i),\sum_{j=1}^{n}a_j K(\cdot,x_j)\rangle \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}a_i a_j\langle K(\cdot,x_i),K(\cdot,x_j)\rangle \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}a_i a_j K(x_j,x_i) \qquad \text{reproducing property} \end{aligned}$$

Then the representer theorem on the error term says that the problem

$$\text{argmin}_{f\in H}\sum_{i=1}^{n}(y_i-f(x_i))^2+g(\|f\|_H)$$

is equivalent to

$$\begin{aligned} &\text{argmin}_{\alpha\in\mathbb{R}^n}\ \sum_{i=1}^{n}(y_i-\sum_{j=1}^{n}\alpha_j K(x_i,x_j))^2+\lambda\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i K(x_i,x_j)\alpha_j \\ &\text{argmin}_{\alpha\in\mathbb{R}}\ \|y-A\alpha\|_2^2+\lambda\alpha^T A\alpha \\ &\text{argmin}_{\alpha\in\mathbb{R}}\left[y^T y-2y^T A\alpha+\alpha^T A^T A\alpha+\lambda\alpha^T A\alpha\right] \\ &\text{argmin}_{\alpha\in\mathbb{R}}\left[\alpha^T\left(A^T A+\lambda A\right)\alpha-2y^T A\alpha\right] \end{aligned}$$

This is convex by positivity, so by differentiating in α, we see that the minimizing choice α^* is given by solving

$$(A^T A + \lambda A)\alpha^* = Ay.$$

Note that while the Representer theorem does not guarantee existence of a minimizer, we can use the finite dimensional representation as a template which may allow us to find a minimizer in an easier way. In this least squares example, it is simple to see existence and uniqueness of the solution.

Example 0.8. Say we have samples $x_1, \dots, x_n \in [0,1]$ and labels $y_1, \dots, y_n \in \mathbb{R}$. We want to design a Hilbert space with norm

$$\|f\|_{\mathcal{H}}^2 = \int_0^1 f'(x)^2 \, dx.$$

Think of this as a measure of complexity.

A first attempt may take the Sobolev space $\mathcal{H} = H^1([0,1])$ consisting of weakly differentiable functions with a square-integrable derivative. However, we can not use the above norm (for instance, any constant function would have zero norm).

Another attempt is to take $\mathcal{H} = \{f \in H^1([0,1]) \mid f(0) = 0\}$ with inner product

$$\langle f, g \rangle = \int_0^1 f'(x) g'(x) \, dx.$$

By the fundamental theorem of calculus, we have

$$f(x) = \int_0^1 G(t,x) f'(t) \, dt$$

where

$$G(t,x) = \begin{cases} 1 & t < x \\ 0 & t > x \end{cases}$$

Since we want the reproducing property $f(x) = \langle f, K(\cdot, x) \rangle$, we take

$$K(t,x) = \begin{cases} t & t < x \\ x & x \geq t \end{cases}$$

These are ramp functions. Say we are looking for a minimizer f^* to a loss

$$f^* = \operatorname{argmin}_{f \in \mathcal{H}} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|_{\mathcal{H}}^2.$$

Then by the Representer theorem f^* takes the form $\sum_{i=1}^n a_i K(x, x_i)$, which is piecewise linear with (possible) knots at $x_1, \dots, x_n$.

Example 0.9 (B-splines)**.** Now, instead of by penalizing complexity by gradients, we penalize by curvature. We want a norm

$$\|f\|_{\mathcal{H}}^2 = \int_0^1 f''(x)^2 \, dx$$

so we want an inner product

$$\langle f, g \rangle = \int_0^1 f''(x) g''(x) \, dx.$$

We can take $\mathcal{H} = \{f \in \mathcal{H}^2([0,1]) \mid f(0) = f'(0) = 0\}$. Then we can derive

$$\begin{aligned} f(x) &= \int_0^x f'(t)\, dt \\ &= \int_0^x \int_0^t f''(s)\, ds\, dt \\ &= \int_0^1 (x-t)_+ f''(t)\, dt \end{aligned}$$

Now, to find the kernel, we can find a g with $g'' = (x-t)_+$ that also satifies $g(0) = g'(0) = 0$. Solving, we have

$$K(t,x) = \begin{cases} \frac{xt^2}{2} - \frac{t^3}{6} & t < x \\ -\frac{x^3}{6} + \frac{x^2 t}{2} & t \geq x \end{cases}$$

Now we have that a minimizer $f^* = \sum_{i=1}^n a_i K(x, x_i)$ is a piecewise cubic function with (possible) knots at $x_1, \ldots, x_n$.

Building more complicated RKHS

Let $K, K_1, K_2 : \Omega \times \Omega \to \mathbb{R}$ be positive definite kernels where $K(x,y) = K_1(x,y) + K_2(x,y)$. Let $\mathcal{H}, \mathcal{H}_1, \mathcal{H}_2$ be the corresponding RKHS's. Then $\mathcal{H} = \mathcal{H}_1 + \mathcal{H}_2$ and

$$\|f\|_{\mathcal{H}}^2 = \min_{\substack{f_1 \in \mathcal{H}_1 \\ f_2 \in \mathcal{H}_2 \\ f_1 + f_2 = f}} \|f_1\|_{\mathcal{H}_1}^2 + \|f_2\|_{\mathcal{H}_2}^2$$

Example 0.10. Let $\mathcal{H}_1 = \{f \in \mathcal{H}^1 \mid f(0) = 0\}$, and say want a Hilbert space that also contains all constant functions. Let $\mathcal{H}_2 = \text{span}\{1\}$. Then we have kernels

$$K_1(t,x) = \begin{cases} t & t < x \\ x & t > x \end{cases} \qquad K_2(t,x) = 1$$

Lecture 10: Applications of RKHS (2/20)

Example 0.11 (Noisy functions)**.** Say we have

- $x_1, \ldots, x_n \in [0,1]$
- $y_1, \ldots, y_n \in \mathbb{R}, \quad y_i = f(x_i) + \epsilon_i$, ϵ_i is random noise

when there is noise, we do not just want to interpolate to determine f, as noise at one point changes the whole interpolant. Instead we solve a problem of the form

$$f^* = \text{argmin}_{f \in \mathcal{H}_1 + \mathcal{H}_2} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \int_0^1 \left| f'(x) \right|^2 dx.$$

The term on the right penalizes the derivative of f. Here we take the space

$$\mathcal{H}_1 = \{f \in \mathcal{H}^1([0,1]) \mid f'(x) \in L^2([0,1]),\ f(0) = 0\}$$

$$\langle f, g\rangle_{\mathcal{H}_1} = \int_0^1 f'(x)g'(x)\,dx$$

And we also have the space to allow constants

$$\mathcal{H}_2 = \{f \in \mathcal{H}^1([0,1]) \mid f = c \in \mathbb{R},\ f(0)^2 < \infty\}$$
$$\langle f, g\rangle_{\mathcal{H}_2} = f(0)g(0)$$

The reproducing kernel is $K_2(t,x) = 1$, since $f(0) = f(x) = \langle f, K_2(\cdot, x)\rangle_{\mathcal{H}_2} = f(0)K(0,x)$. Thus, $\mathcal{H} = \mathcal{H}_1 + \mathcal{H}_2$ has reproducing kernel $1 + K_1(t,x)$.

Going through the proof of the Representer Theorem, it can be shown that $f^* = c + \sum_{i=1}^n a_i K_1(x, x_i) = \tilde{c} + \sum_{i=1}^n b_i (x - x_i)_+$. This is piecewise linear, and we can plug it into the error function to solve for the coefficients.

Example 0.12 (Support vector machine)**.** Say we have data for a classification problem

- $x_1, \ldots, x_n \in \mathbb{R}^d$.
- $y_1, \ldots, y_n \in \{-1, 1\}$.

The goal is to find a separating hyperplane $f(x) = w^T x + c$ where $y_i f(x_i) > 0$. Then the classifier would be to predict $\text{sign}(w^T x + c)$ for point x. Moreover, we want to find such a hyperplane with maximum margin, which takes the form $\frac{2}{\|w\|}$. Thus, our objective is

$$\min \|w\|_2 \quad \text{s.t. } y_i(w^T x_i + c) > 0,\ i = 1, \ldots, n$$

These constraints are not generally feasible, so instead we consider soft-margin SVM. Here we let the error function be a hinge loss

$$E\left((x_1, y_1, f(x_1)), \ldots, (x_n, y_n, f(x_n))\right) = \sum_{i=1}^n \max(0, 1 - y_i f(x_i))$$

The objective is

$$f^* = \text{argmin}_{f \in \mathcal{H}_1 + \mathcal{H}_2} \sum_{i=1}^n \max(0, 1 - y_i f(x_i)) + \lambda \|f\|_{\mathcal{H}_1}^2$$

where the spaces are $\mathcal{H}_1 = \{x \mapsto w^T x \mid w \in \mathbb{R}^d\}$. If $f(x) = w^T x$, then $\|f\|_{\mathcal{H}_1}^2 = \|w\|_2^2$, and the inner product is $\langle f, g\rangle_{\mathcal{H}_1} = \langle x \mapsto w_1^T x, x \mapsto w_2^T x\rangle_{\mathcal{H}_1} = w_1^T w_2$. The reproducing kernel takes the form

$$\begin{aligned} w^T x &= f(x) \\ &= \langle \underbrace{f}_{w^T x}, \underbrace{K_1(\cdot, x)}_{\mu_x^T x}\rangle_{\mathcal{H}_1} \\ &= w^T \mu_x \\ \implies K(y, x) &= y^T x \end{aligned}$$

The other space consists of translations (constant functions) $\mathcal{H}_2 = \{c \mid c \in \mathbb{R}\}$, with the inner product and kernel as defined above.

For the objective where the regularizer is replaced by $\lambda \|f\|_{\mathcal{H}}^2$, we have a minimizer by the representer theorem of the form:

$$g^*(x) = \sum_{i=1}^n a_i (1 + x^T x_i) = \sum_{i=1}^n a_i + \left(\sum_{i=1}^n a_i x_i\right)^T x$$

Thus, the solution to the original problem takes $f^*(x) = c + \left(\sum_{i=1}^n a_i x_i\right)^T x$.

RKHS and Neural Nets

Say we have a neural net with activation functions $\sigma : \mathbb{R} \to \mathbb{R}$. In 2019, Michael Unser considered an optimization problem to choose activation functions. Say we have data $x_1, \dots, x_n \in \mathbb{R}$ and $y_1, \dots, y_n \in \mathbb{R}$. We need the activation functions to satisfy

- σ should be (weakly) differentiable
- σ should have a weak second derivative that is sparsely supported

We define the space $\mathcal{H}_1 = \{f \in BV^{(2)}(\mathbb{R}) \mid f(0) = f'(0) = 0\}$, where $BV^{(2)}(\mathbb{R})$ is the space of functions with some constraint of bounded variation on their second derivative. The norm is $\|f\|_{BV^{(2)}}$, which penalizes large second derivatives. Also, $\mathcal{H}_2 = \{f \in BV^{(2)}(\mathbb{R}) \mid f(x) = a + bx\}$, with the norm $\|f\|_{\mathcal{H}_2}^2 = a^2 + b^2$. For each activation function, we have the optimization problem

$$\sigma^* = \text{argmin}_{\sigma \in BV^{(2)}(\mathbb{R})} E\left((x_1, y_1, f(x_1)), \dots, (x_n, y_n, f(x_n))\right) + \lambda \|f\|_{BV^{(2)}}$$

Unser shows that the minimizer takes the form

$$\sigma^* = b_1 + b_2 x + \sum_{i=1}^{n} a_i (x - x_i)_+$$

which is basically a linear combination of ReLU activations, plus a linear term. Also, if $b_1 = b_2 = 0$, then $\|\sigma^*\|_{BV^{(2)}} = \sum_{i=1}^n a_i$. This penalizes the number of pieces in the linear combination of ReLUs.

Lecture 11: Linear Algebra for Functions (2/27)

We can define a linear algebra for functions that is mostly analogous to linear algebra for matrices.

Linear algebra for matrices	Linear algebra for functions
Matrix: $A : [n] \times [n] \to \mathbb{C}$	Function: $K : \Omega \times \Omega \to \mathbb{C}$
Element: $A(j,k) = A_{jk}$	Value: $K(x,y)$
Mat-vec: $(Av)_j = \sum_{k=1}^n A_{jk} v_k$	Integral operator: $w(x) = \int_\Omega K(x,y) v(y)\, dy$
2-norm: $\|A\|_2 = \max_{v \neq 0} \frac{\|Av\|_2}{\|v\|_2}$	Operator norm: $\|T_k\|_{\text{op}} \sup_{v \in L^2(\Omega)} = \frac{\|T_k v\|_{L^2}}{\|v\|_{L^2}}$ for $T_k : L^2(\Omega) \to L^2(\Omega)$
Frob-norm: $\|A\|_F^2 = \sum_{i,j}^n \left\|A_{ij}\right\|^2$	L^2-norm: $\|K\|_{L^2}^2 = \int_\Omega \int_\Omega \left\|K(x,y)\right\|^2\, dx\, dy$

Where we define $T_k v(y) = \int_\Omega K(x,y) v(x)\, dx$. Note that this is bounded if $\|K\|_{L^2} < \infty$, by Cauchy-Schwarz, since

$$\begin{aligned}
\|T_k v\|_{L^2} &= \int_\Omega \left|T_k v(y)\right|^2\, dy \\
&= \int_\Omega \left| \int_\Omega K(x,y) v(x)\, dx \right|^2\, dy \\
&\leq \int_\Omega \int_\Omega \left|K(x,y)\right|^2 \int_\Omega \left|v(x)\right|^2\, dx\, dy
\end{aligned}$$

Much as we often decompose matrices into simpler matrices, we can decompose functions into simpler functions. We will often need additional assumptions/ restrictions on our functions to be able to do this.

We can define matrices as functions on a domain $[m] \times [n]$, and quasimatrix as a function on $[a,b] \times [n]$, and a cmatrix as a function on $[a,b] \times [c,d]$. Algorithms with functions on a domain $[m]$ are much easier than dealing with functions on a domain $[a,b]$. For instance, in the latter case we have to deal with

- No successor. We can deal with this by pivoting.
- Null subsets. We can deal with this by restricting to smooth functions.
- Convergence. We can deal with this by truncating at ϵ_{mach}.

SVD

Recall the SVD, $A = U\Sigma V^T$. It has the properties

- Existence: The SVD exists and is almost unique.
- Application: A best rank r approximation to A in the Frobenius norm is given by a rank r truncation of the SVD.
- Separable model: $A = \sum_{j=1}^{n} \sigma_j u_j v_j^T$ is a sum of outer products.
- Computation: Lots of SVD algorithms. Many rely on bidiagonalization.

We can define a (formally) continous analogue $A = U\Sigma V^T$, where U, V have orthogonal columns, and Σ is diagonal. In this case, U, V are quasimatrices, and Σ is an infinite diagonal matrix.

- Existence: If A is integrable, then the SVD exists and is almost unique (Schmidt 1907).
- Application: A best rank r approximation in the L^2 norm is given by the first r terms of the SVD (Weyl 1912).
- Separable model: $A = \sum_{j=1}^{\infty} \sigma_j u_j v_j^T$ is a sum of "outer products".
- Computation: Avoid bidiagonalization.

Theorem 17. *Let A be an $[a,b] \times [c,d]$ cmatrix that is uniformly Lipschitz continuous in both variables. Then the SVD of A exists, the singular values are unique with $\sigma_j \to 0$ as $j \to \infty$, and*

$$A = \sum_{j=1}^{\infty} \sigma_j u_j v_j^T$$

where the series is uniformly and absolutely convergent to A.

LU

We can decompose a matrix $A = P^{-1}LU$.

- Existence: Almost exists and with extra conditions is almost unique.

- Application: Used to solve dense linear systems $Ax = b$.
- Separable model: $A = \sum_{j=1}^{n} l_j u_j^T$ is a sum of outer products (Pan 2000).
- Computation: Gaussian elimination with pivoting.

For a cmatrix, we can decompose $A = LU$, where L is unit lower triangular and U is upper triangular, which we will define.

- Usually exists and with extra conditions is almost unique.
- Can be used to "solve" integral equations.
- $A = \sum_{j=1}^{\infty} l_j u_j^T$ is a sum of outer products.
- Continuous analogue of Gaussian elimination with complete pivoting.

We have to define triangularity of a quasimatrix. A lower-triangular matrix has at least $j-1$ zeros in its jth column, and has a nesting structure, where $j-1$ zero rows on column j should have corresponding zero rows on column $j-1$. Taking this analogously with a quasimatrix with discrete columns gives our definition.

Cholesky

For A a positive definite matrix, $A = R^T R$ for R upper triangular.

- Existence: Exists and is unique if A is pos def.
- Application: A numerical test for a positive definite matrix.
- Separable Model: $A = \sum_{j=1}^{n} r_j r_j^T$
- Algorithm: Cholesky algorithm (GECP on a pos def matrix).

We can define a nonnegative definite cmatrix A on $[a,b] \times [a,b]$ as one that is continuous, symmetric, and

$$v^T A v = \int_a^b \int_a^b v(y) A(y,x) v(x)\, dx\, dy \geq 0 \qquad \forall v \in C[a,b]$$

Hilbert-Schmidt integral operators

Given a domain $\Omega \subseteq \mathbb{R}^d$, a Hilbert-Schmidt kernel has $\|K\|_{L^2(\Omega)} < \infty$. Then

$$T_k v(y) = \int_\Omega K(x,y) v(x)\, dx$$

is a Hilbert-Scmidt integral operator.

Lecture 12: Singular Value Decomposition of Functions (3/3)

We define the singular values and vectors of a Hilbert-Schmidt operator T_K, where $K : [a,b] \times [c,d] \to \mathbb{C}$ is square integrable, as

$$\sigma_1(K) = \sigma_1(T_K) = \sup_{\substack{v \in L^2[a,b] \\ v \neq 0}} \frac{\sqrt{\int_c^d \int_a^b |K(x,y)v(y)\,dx|^2\,dy}}{\sqrt{\int_a^b |v(x)|^2\,dx}}$$

The maximum is attained, since we can scale $\|v\|_{L^2} = 1$ and K is square-integrable, so we can apply Cauchy-Schwarz $\|T_K v\|_{L^2} = \|K\|_{L^2}\|v\|_{L^2}$. Call an argmin with unit norm $v_1(x)$. This is the first right singular vector. Let the left singular vector $u_1(y)$ be defined as $u_1 = \frac{1}{\sigma_1(K)} T_K v_1$.

Then we recursively define $\sigma_1, \ldots, u_1, \ldots, v_1, \ldots$ by first defining $K_{j-1}(x,y) = \sum_{s=1}^{j-1} \sigma_s(K) u_s(y) \overline{v_s(x)}$. Then we compute the jth singular values and vectors by using

$$\sigma_j(K) = \sigma_1(K - K_{j-1}).$$

It can be checked that the right singular vectors are orthonormal and also the left singular vectors are orthonormal.

Theorem 18. *Let $K : [a,b] \times [c,d] \to \mathbb{C}$ be Lipschitz continuous in both variables. Then the SVD exists, the singular values are unique and $\sigma_j \to 0$. The singular functions are continuous, and*

$$K(x,y) = \sum_{j=1}^{\infty} \sigma_s(K) u_s(y) \overline{v_s(x)}$$

which converges absolutely and uniformly.

Lipschitz continuity is in fact a little stronger than needed for this result, some bounded variation condition is sufficient.

Definition 0.4. A continuous function $K : [a,b] \times [c,d] \to \mathbb{C}$ is of rank $\leq R$ if $K(x,y) = \sum_{j=1}^{R} c_j(y) \overline{r_j(x)}$.

Lemma 1. *Let $K : [a,b] \times [c,d] \to \mathbb{C}$ be Lipschitz in both variables. If K_R is Lipschitz and of rank $\leq R$, then*

$$\sigma_1(K - K_R) = \|T_{K-K_R}\|_{\text{op}} \geq \sigma_{r+1}(K)$$

Proof. Let $K(x,y) = \sum_{j=1}^{\infty} \sigma_j(K) u_j(y) \overline{v_j(x)}$ and $K_R(x,y) = \sum_{j=1}^{R} c_j(y) \overline{r_j(x)}$. Since K_R has rank $\leq R$,

$$\int_a^b K_R(x,y) w(x)\,dx = 0$$

for some $w(x) = \sum_{j=1}^{R+1} \gamma_j v_j(x)$ in the nullspace. We can pick this such that $\gamma_1^2 + \ldots + \gamma_{R+1}^2 = 1$.

Then

$$\begin{aligned}
\|T_{K-K_R}\|_{\text{op}}^2 &\geq \|T_{K-K_R}w\|_{L^2}^2 \\
&= \left\| \int_a^b (K(x,\cdot) - K_R(x,\cdot))w(x)\,dx \right\|_{L^2}^2 \\
&= \left\| \int_a^b K(x,\cdot)w(x)\,dx \right\|_{L^2}^2 \\
&= \|T_K w\|_{L^2}^2 \\
&= \sum_{j=1}^{R+1} \gamma_j^2 \sigma_j(K)^2 \\
&\geq \sigma_{R+1}(K)^2
\end{aligned}$$

□

A similar argument shows that $\sigma_j(K - K_R) \geq \sigma_{j+R}(K)$, for $j \geq 1$.

Thus, we have that

$$\begin{aligned}
\|K - K_R\|_{L^2}^2 &= \sum_{j=1}^{\infty} \sigma_j(K - K_R)^2 \\
&\geq \sum_{j=1}^{\infty} \sigma_{j+R}(K)^2 \\
&= \sum_{j=R+1}^{\infty} \sigma_j(K)^2
\end{aligned}$$

Thus, K_R is the best rank R truncation of the SVD of K in this sense. Now, to show the effectiveness of this truncation, we wish to show that the singular values decay quickly. To do this, we bound the error of other approximations.

Recall that if $f : [-1,1] \to \mathbb{C}$ and $f, \ldots, f^{(v-1)}$ are absolutely continuous and $f^{(v)}$ is of bounded variation $V < \infty$, then

$$\|f - p_n\|_\infty \leq \frac{4V}{\pi v (n-v)^v}, \quad n > v$$

Let $K : [-1,1] \times [-1,1] \to \mathbb{C}$, so $K(\cdot, y)$ satisfies $(*)$ uniformly for $y \in [-1,1]$. Then

$$\left\| K(\cdot,y) - \sum_{s=0}^{j-2} c_s(y) T_s(\cdot) \right\|_\infty \leq \frac{4V}{\pi v (n-v-2)^v}, \quad j - 2 > v.$$

Now we have that

$$\begin{aligned}
\sigma_j(K) &\le \sqrt{\sum_{s=j}^{\infty} \sigma_s(K)^2} \\
&= \|K - K_{j-1}\|_{L^2} && K_{j-1} \text{ is truncated SVD} \\
&\le \left\| K - \sum_{s=0}^{j-2} c_s(y) T_s(\cdot) \right\|_{L^2} && \text{optimality of truncated SVD} \\
&\le 4 \sup_{y \in [-1,1]} \left\| K(\cdot, y) - \sum_{s=0}^{j-2} c_s(y) T_s(\cdot) \right\|_{\infty} \\
&\le \frac{16V}{\pi v (j - v - 2)^v}
\end{aligned}$$

Lecture 13: LU and Cholesky for Functions (3/5)

Recall that for an $m \times n$ matrix A of rank r, it can be written as $A = LU$ for an $m \times r$ psychologically lower triangular matrix L and an $r \times n$ psychologically upper triangular matrix U. We say that a matrix $L \in \mathbb{C}^{m \times r}$ is psychologically lower triangular if there are numbers $i_1, \dots, i_r$ such that

$$L(i_j, s+1) = 0, \quad 1 \le j \le s,\ 1 \le s \le r-1.$$

This means that column $s+1$, there are s zeros, and at least $s-1$ are nested with the zeros of column s. We write it in this way so that it is easily generalizable to functions.

Also, we can write the Gaussian elimination algorithm

$$\begin{aligned}
&A_1 = A \\
&A_{s+1} = A_s - \tfrac{A_s(:, j_s) A_s(i_s, :)}{A_s(i_s, j_s)} && \text{zeros out the } i_s \text{ row and } j_s \text{ column}
\end{aligned}$$

For instance, after two steps, we have $A_1 = \begin{bmatrix} \frac{A_1(:,j_1)}{A_s(i_1,j_1)} & \frac{A_2(:,j_2)}{A_2(i_2,j_2)} \end{bmatrix} \begin{bmatrix} A_1(i_1,:) \\ A_2(i_2,:) \end{bmatrix} + A_3$. Note that zeros are introduced and they are preserved along indices in the rank one factors making up the left summand.

The same algorithm works for continuous functions $K : [a,b] \times [c,d] \to \mathbb{C}$.

$$\begin{aligned}
&K_1 = K \\
&K_{s+1}(x,y) = K_s(x,y) - \tfrac{K_s(x_s,y) K_s(x,y_s)}{K_s(x_s,y_s)} \\
&l_s(y) = \tfrac{K_s(x_s,y)}{K_s(x_s,y_s)}, \quad u_s(x) = K_s(x,y_s)
\end{aligned}$$

Then we can write

$$L = \begin{bmatrix} l_1(y) & l_2(y) & \dots \end{bmatrix}, \ U = \begin{bmatrix} u_1(x) \\ u_2(x) \\ \vdots \end{bmatrix}$$

The algorithm can be written

$$\begin{aligned} K(x,y) &= l_1(y)u_1(x) + K_2(x,y) \\ &= l_1(y)u_1(x) + l_2(y)u_2(x) + K_3(x,y) \\ &= \sum_{j=1}^{\infty} l_j(y)u_j(x) \\ &= LU \end{aligned}$$

where we will later discuss convergence of the series. Both L and U are psychologically triangular.

There are multiple ways to choose pivots. In interpolative decomposition, there is randomness in the choice of pivots. Pseudoskeleton methods solve some optimization problem to choose the best pivots. In adaptive cross approximation (ACA) pivots are often selected to give the absolute maximum of $K_s(x,y)$ at step s.

One way to compute the rank r SVD is:

Run r steps of ACA to get $l_1(y),\ldots,l_r(y)$ and $u_1(x),\ldots,u_r(x)$

Factor L as $Q_L R_L$ and factor $U^* = Q_U R_U$. This can be done by Householder transforms

Then $K \approx LU = Q_L R_L R_U^* Q_U^*$.

$R_L R_U^*$ is $r \times r$, so do an SVD $R_L R_U^* = U_R \Sigma_R V_R^*$

$K \approx LU = (Q_L U_R)\Sigma_R (V_R^* Q_U^*)$

It can be checked that $Q_L U_R$ has orthogonal columns in the LU sense. To make this better, oversample by running more than r of ACA to get a better LU approximation to K. This works well in practice, but there are no theoretical guarantees on it (as we do not know the stability and convergence of Gaussian elimination in general).

Cholesky Factorization

Say that K is nonnegative definite now. We can factor $K = LU = R^*R$, by scaling so that $L^* = U$. If A is positive definite, then the absolute maximum is on the diagonal and is a positive element. Likewise, for K nonnegative definite, the absolute maximum is on the diagonal and is a positive element. This makes Gaussian elimination with complete pivoting much easier. Also, it gives a way to check that kernels are positive definite—simply run Gaussian elimination with complete pivoting, and if there is every a nondiagonal pivot, the kernel is not positive definite. Thus, GECP takes the form

Pivots $(x_1,x_1),\ldots,(x_r,x_r),\ldots$

$l_j(y) = \frac{K_s(x_s,y)}{\sqrt{K_s(x_s,x_s)}}, \quad u_s(x) = \frac{K_s(x,x_s)}{\sqrt{K_s(x_s,x_s)}}$

Lecture 14: Gaussian Processes and Linear Algebra on Functions (3/10)

Let Ω be a nonempty set, $K : \Omega \times \Omega \to \mathbb{R}$ a positive definite kernel (analogous to covariance), $m : \Omega \to \mathbb{R}$ a function (analogous to mean). The random function $f : \Omega \to \mathbb{R}$ is a Gaussian process $GP(m, k)$ if

- For any finite set of points $x = (x_1, \ldots, x_n)$, $x_i \in \Omega$, the random vector $f_x = (f(x_1), \ldots, f(x_n))^T$ follows a multivariate Gaussian distribution $\mathcal{N}(m_x, K_{xx})$, where

$$m_x = (m(x_1), \ldots, m(x_n)), \quad K_{xx} = \begin{bmatrix} K(x_1, x_1) & \ldots & K(x_1, x_n) \\ \vdots & \ddots & \vdots \\ K(x_n, x_1) & \ldots & K(x_n, x_n) \end{bmatrix}$$

Oftentimes, the kernel is chosen to be a Gaussian kernel $K(x, y) = e^{-\frac{\|x-y\|^2}{2\gamma^2}}$, $\gamma > 0$.

Discrete Sampling

In the discrete case, we consider how to sample $GP(0, K_{xx})$, where $\Omega = X$ is a finite set. The probability density of a point is

$$p(u) = \frac{1}{(2\pi)^{n/2}} \frac{1}{\det(K_{xx})^{1/2}} e^{-\frac{1}{2} u^T K_{xx}^{-1} u}$$

The algorithm for sampling is:

Compute $(K_{xx})_{ij} = K(x_i, x_j)$

$u = \texttt{randn}(\texttt{n}, 1)$, so $u \sim \mathcal{N}(0, I_n)$

Take a square root (e.g. Cholesky) $K_{xx} = A^T A$, then

$$e^{-\frac{1}{2} u^T (A^T A)^{-1} u} = e^{-\frac{1}{2} u^T A^{-1} A^{-T} u} = e^{-\frac{1}{2} (A^{-T} u)^T (A^{-T} u)}$$

The sample is $f = A^T u$

To sample from $GP(m, K)$, just take $u - m$ as u, so $f = A^T u + m$.

Continuous Sampling

We can also use linear algebra for functions to sample from a $GP(0, K)$ where Ω is infinite.

Compute an SVD $K = U \Sigma V^*$ (Since $K^T = K$, $U = V$, and $K = (U \Sigma^{1/2})(\Sigma^{1/2} U^*)$)

$u = \texttt{randn}(\infty, 1)$

$f = U \Sigma^{1/2} u$

To make this pratical, let r be a rank where σ_{r+j} is below machine precision for $j > 0$. Then we truncate the SVD at r columns, and sample `randn(r, 1)`.

Theorem 19 (Mercer)**.** *Let $K : \Omega \times \Omega \to \mathbb{R}$ be a continuous symmetric positive definite kernel, where $\Omega \subseteq \mathbb{R}^d$ is compact. Then*

$$K(x,y) = \sum_{j=1}^{\infty} \lambda_j e_j(y) e_j(x)$$

where the series converges absolutely and uniformly, and $(T_k e_j)(y) = \int_\Omega K(x,y) e_j(x)\, dx = \lambda e_j(y)$.

Gaussian processes and kernel-based learning

The task is to learn $g : \Omega \to \mathbb{R}$ given training data of the form

$$y_i = g(x_i) + \xi_i, \quad 1 \le i \le n$$

where ξ_i are zero-mean random variables. We will set up the problem as

$$\hat{h} = \operatorname{argmin}_{h \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^{n} (y_i - h(x_i))^2 + \lambda \|h\|_{\mathcal{H}_K}^2$$

where $\mathcal{H}_K$ is an RKHS with norm that penalizes derivatives. Then the representer theorem says that $\hat{h}(x) = \sum_{i=1}^n \alpha_i K(x, x_i)$. The solution is $(K_{xx} + \lambda I)\alpha = y$.

Learning with Gaussian processes takes a Bayesian approach, so we have:

- Prior distribution Π_0 on g
- Likelihood function $l_{x,y}(h)$
- Posterior distribution Π_n

Gaussin process regression takes the form

- Prior Π_0 is a Gaussian process $GP(m, K)$
- Likelihood function is normal $l_{x,y}(h) = \prod_{i=1}^n \mathcal{N}(y_i \mid h(x_i), \sigma) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}} e^{-\frac{(y_i - h(x_i))^2}{2\sigma^2}}$
- Abusing notation, $d\Pi_n(h \mid x, y) \propto l_{x,y}(h) d\Pi_0(h)$
- It can be shown $d\Pi_n \sim GP(\hat{m}, \hat{K})$, where

$$\hat{m}(x) = m(x) + \begin{bmatrix} K(x,x_1) & \dots & K(x,x_n) \end{bmatrix} (K_{xx} + \sigma^2 I_n)^{-1} \begin{bmatrix} y_1 - m(x_1) \\ \vdots \\ y_n - m(x_n) \end{bmatrix}$$

$$\hat{K}(x,y) = K(x,y) - \begin{bmatrix} K(x,x_1) & \dots & K(x,x_n) \end{bmatrix} (K_{xx} + \sigma^2 I_n)^{-1} \begin{bmatrix} K(x_1,y) \\ \vdots \\ K(x_n,y) \end{bmatrix}$$

Note that the covariance update is like a Schur complement, so it is like taking n Gaussian elimination steps.

Lecture 15: (3/12)

Again, we wish to learn $g : \Omega \to \mathbb{R}$ given training data of the form

$$y_i = g(x_i) + \xi_i, \quad 1 \le i \le n$$

where ξ_i are random variables with mean zero and variance σ^2. For GP regression, we model

- Prior $g \sim GP(m, K)$
- Observations $(x_1, y_1), \ldots, (x_n, y_n)$
- Posterior $g \sim GP(\hat{m}, \hat{K})$

where $\hat{m}$ and $\hat{K}$ are updated according to the formulas in the last lecture, which is like n Gaussian elimination/ Cholesky updates on K. At the sample points, $\hat{K}$ is thus zeroed out, so there is no variance of predictions at the sample points.

KL expanasion

Theorem 20 (Spectral theorem)**.** *Let $K : \Omega \times \Omega \to \mathbb{R}$ be symmetric and square integrable $\|K\|_{L^2(\Omega\times\Omega)} < \infty$. Consider*

$$(T_k w)(y) = \int_\Omega K(x, y) w(x)\, dx$$

then there are countably many eigenvalues $\lambda_1, \lambda_2, \ldots$, so $(Te_j)(y) = \lambda_j e_j(y)$ where $e_1, e_2, \ldots$ is an orthonormal basis of $L^2(\Omega)$.

Recall that if K is positive definite and continuous, then Mercer's theorem says that

$$K(x, y) = \sum_{j=1}^{\infty} \lambda_j e_j(y) e_j(x).$$

Now, let $\mathcal{H}_K$ be the RKHS of this kernel. We want to connect GPs to RKHS, so that we can use the strong tools like approximation theory.

The reproducing property says

$$\begin{aligned} e_i(x) &= \langle e_i, K(\cdot, x)\rangle_{\mathcal{H}_K} \\ &= \langle e_i, \sum_{j=1}^{\infty} \lambda_j e_j e_j(x)\rangle_{\mathcal{H}_K} \\ &= \sum_{j=1}^{\infty} \lambda_j e_j(x) \langle e_i, e_j\rangle_{\mathcal{H}_K} \\ \implies \lambda_j \langle e_i, e_j\rangle_{\mathcal{H}_K} &= \begin{cases} 1 & i = j \\ 0 & i \ne j \end{cases} \end{aligned}$$

since the e_j form a basis. Note that $\lambda_j \langle e_i, e_j\rangle_{\mathcal{H}_K} = \langle \lambda_j^{1/2} e_i, \lambda_j^{1/2} e_j\rangle_{\mathcal{H}_K}$. Thus, $\{\lambda_j^{1/2} e_j\}$ is an orthonormal basis for $\mathcal{H}_K$. Thus, we have Mercer's representation,

$$\mathcal{H}_K = \left\{ f = \sum_{i=1}^{\infty} \alpha_i \lambda_i^{1/2} e_i \mid \|f\|_{\mathcal{H}_K}^2 = \sum_{i=1}^{\infty} \alpha_i^2 < \infty \right\}$$

$$\langle f,g\rangle_{\mathcal{H}_K} = \sum_{i=1}^{\infty} \alpha_i\beta_i, \text{ where } f = \sum_{i=1}^{\infty} \alpha_i\lambda_i^{1/2} e_i,\ g = \sum_{i=1}^{\infty} \beta_i\lambda_i^{1/2} e_i$$

Now, let us go back to Gaussin processes. View $GP(m,K)$ as a collection of random variables X_t, $t \in [a,b]$, where

$$\mathbb{E}[X_t] = m(t), \qquad \mathbb{E}[X_s, X_t] = K(s,t)$$

The following theorem gives a Fourier-like decomposition of the GP.

Theorem 21 (KL expansion)**.** *Let $GP(0,K)$ be a Gaussian process, where $K; [a,b]\times[a,b] \to \mathbb{R}$ is continuous. Then*

$$X_t = \sum_{j=1}^{\infty} Z_j e_j(t)$$

in which e_j are eigenfunctions of T_K and $Z_j = \int_a^b X_t e_j(t)\,dt$ are zero-mean and uncorrelated.

Proof. We compute that

$$\begin{aligned}
\mathbb{E}[Z_j] &= \mathbb{E}\left[\int_a^b X_t e_j(t)\,dt\right] \\
&= \int_a^b \mathbb{E}[X_t] e_j(t) \\
&= 0 \\
\mathbb{E}[Z_i Z_j] &= \mathbb{E}\left[\int_a^b \int_a^b X_t X_s e_j(t) e_i(s)\,dt\,ds\right] \\
&= \int_a^b \int_a^b \mathbb{E}[X_t X_s] e_j(t) e_i(s)\,dt\,ds \\
&= \int_a^b \int_a^b K(s,t) e_j(t) e_i(s)\,dt\,ds \\
&= \lambda_i \int_a^b e_j(t) e_i(t)\,dt \\
&= \begin{cases} \lambda_k & i = j \\ 0 & i \neq j \end{cases}
\end{aligned}$$

where we can apply Fubini in those two cases due to boundedness (Cauchy-Schwarz). □

In fact, it can be shown that the Z_j are Gaussian distributed. This expansion is useful for analyzing Gaussian Processes.

Now, we present something that says that approximation theory for Gaussian processes is difficult.

Theorem 22 (Driscoll's Zero-One Law)**.** *Let $f \sim GP(0,K)$.*

Zero: *With probability zero, $f \in \mathcal{H}_K$.*

One: *With probability one, $f \in \mathcal{H}_K^\theta$, for any $0 < \theta < 1$, where*

$$H_K^\theta = \left\{ f = \sum_{i=1}^{\infty} \alpha_i \lambda_i^{\theta/2} e_i \mid \sum_{i=1}^{\infty} \alpha_i^2 < \infty \right\}$$

Lecture 16: (4/7)

First lecture back from coronavirus disruption

Schmidt's integral theorem

Recall that we say that a bounded linear operator $T : L^2(\Omega) \to L^2(\Omega)$ is a Hilbert-Schmidt operator if

$$\|T\|_{HS}^2 = \sum_{j=1}^{\infty} \sigma_j(T)^2 < \infty$$

Schmidt's integral theorem says that such an operator can be written as an integral operator (just as linear maps in finite dimensional spaces can be written as matrix-vector products). This is an extremely useful way to study operators, as we then have many tools to use, such as the decompositions that we have studied earlier.

Theorem 23 (Schmidt's integral theorem)**.** *If $T : L^2(\Omega) \to L^2(\Omega)$ is a Hilbert-Schmidt operator, then there is a square-integrable $K : L^2(\Omega \times \Omega) \to \mathbb{R}$ such that*

$$(Tw)y = \int_\Omega K(x,y)w(x)\,dx$$

Proof. To prove this theorem, we build a kernel. Let $e_1, e_2, \ldots$ be an orthonormal basis for $L^2(\Omega)$. Let $e_{m,n} = e_m(x)e_n(y)$. This is an orthonormal basis for $L^2(\Omega \times \Omega)$. Define

$$K(x,y) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} \langle Te_m, e_n\rangle e_{m,n}(x,y)$$

Then we have that

$$\begin{aligned}
\int K(x,y)w(x)\,dx &= \sum_m \sum_n \langle Te_m, e_n\rangle \int e_{m,n}(x,y)w(x)\,dx \\
&= \sum_m \sum_n \langle Te_m, e_n\rangle e_n(y) \int e_m(x)w(x)\,dx \\
&= \sum_m \langle e_m, w\rangle \sum_n \langle Te_m, e_n\rangle e_n(y) \\
&= \sum_m \langle e_m, w\rangle Te_m(y) \\
&= T\left(\sum_m \langle e_m, w\rangle e_m(y)\right) \\
&= (Tw)(y)
\end{aligned}$$

□

Peetre's theorem

On the flipside, Peetre proved that under certain conditions, some unbounded operators are differential operators. In short any linear, local operators are differential operators.

Definition 0.5 (Univariate differential operators)**.** Let $V \subset \mathbb{R}$ be open. A differential operator on V is a linear map $\mathcal{L} : C_c^\infty(V) \to C_c^\infty(V)$ such that

$$(Lu)(x) = \sum_{j=1}^{N} a_j(x) \frac{d^j u}{dx^j} \qquad \forall u \in C_c^\infty(U),\, U \subset V$$

Such an operator is local in the sense that it does not grow the support of the input. Here we have $\text{supp}(u) = \overline{\{x \in \mathbb{R} : u(x) \neq 0\}}$. Thus, if $\mathcal{L}$ is a differential operator, $\text{supp}(\mathcal{L}u) \subseteq \text{supp}(u)$. Peetre's theorem says that the converse holds as well.

Example 0.13. Define an operator on d-dimensional real functions

$$(\mathcal{L}f)(x_o) = \lim_{r \to 0} \frac{2d}{r^2 |S_r|} \int_{S_r(x_0)} (f(x) - f(x_0))\, dx$$

In one dimension, this is given by

$$(\mathcal{L}f)(x_0) = \lim_{r \to 0} \frac{2}{r^2} \frac{f(x_0 - r) + f(x_0 + r) - 2f(x_0)}{2}$$

Which is the second derivative $f''(x_0)$. In d dimensions, this operator is the Laplacian $\mathcal{L}f(x_0) = \Delta f(x_0)$. Note that this definition is completely coordinate free.

Lecture 17: Peetre's Theorem (4/9)

Here we go over a sketch of a proof for Peetre's theorem. We restrict to Euclidean spaces, but Peetre's theorem is often stated (and applied) for smooth manifolds.

Theorem 24 (Peetre's Theorem)**.** *Let $V \subseteq \mathbb{R}$ be open. If $\mathcal{L} : C_c^\infty(V) \to C_c^\infty(V)$ is a linear, local operator, then $\mathcal{L}$ is a differential operator.*

Proof sketch. Let $u \in C_c^\infty(V)$. We can Taylor expand

$$u(x) = \sum_{k=0}^{N} \frac{u^{(k)}(a)}{k!} (x - a)^k + R_{N,a}(x)$$

Then applying our operator gives

$$\mathcal{L}u = \sum_{k=0}^{N} \frac{u^{(k)}(a)}{k!} \mathcal{L}[(x - a)^k] + \mathcal{L}R_{N,a}$$

Define remainders $\tilde{R}_{N,a}^{(k)}$ of the form

$$\tilde{R}_{N,a}^{(K)}(x) = \begin{cases} R_{N,a}(x) & \text{outside } [a - 2/K, a + 2/K] \\ 0 & \text{inside } [a - 1/K, a + 2/K] \\ \text{smooth interpolation} & \text{elsewhere} \end{cases}$$

Since $\mathcal{L}$ is local, $\mathcal{L}\tilde{R}_{N,a}^{(K)} = 0$ in $[a - 1/K, a + 1/K]$.

The rest of the proof skips some technical details. For U_a an open neighborhood of a,

$$\lim_{K\to 0}\sup_{x\in U_a}\left|\mathcal{L}R_{N,a}(x)-\mathcal{L}\tilde{R}^{(K)}_{N,a}(x)\right|=0$$

so that $\mathcal{L}[R_{N,a}](a)=\tilde{\mathcal{L}}[R_{N,a}](a)$. This means that

$$(\mathcal{L}u)(a)=\sum_{k=0}^{N}\frac{u^{(K)}(a)}{k!}\mathcal{L}[(x-a)^k](a)$$

This is a differential equation at a. We can get a global differential equation that holds on the whole of V by taking a finite subcover and piecing together the individual equations for each a. □

Linear algebra with differential operators

Unlike the nice Hilbert-Schmidt operators, different operators are in general unbounded. For instance, for $\mathcal{L}=\frac{d^2}{dx^2}$, note that $\mathcal{L}\sin(kx)=-k^2\sin(kx)$. The input is $O(1)$ and the output is $O(k^2)$, so letting k grow gets us in trouble.

Example 0.14. Say we are considering a bounded domain $[0,L]$, and consider the eigenvalue equation

$$\begin{aligned}-\frac{d^2u}{dx^2}&=\lambda u\\ u(0)&=u(L)=0\end{aligned}$$

Solutions are given by

$$\begin{aligned}u_j(x)&=\sin\left(\frac{j\pi x}{L}\right)\\ \lambda_j&=\left(\frac{j\pi}{L}\right)^2,\ j\geq 1\end{aligned}$$

And the eigenfunctions u_j form an orthogonal basis for $L^2[0,L]$. The eigenvalues are well-defined, but unbounded.

Example 0.15. Now, say that we are over all of $\mathbb{R}$, so the domain is unbounded. Consider the same eigenvalue equation with no boundary conditions

$$-\frac{d^2u}{dx^2}=\lambda u$$

"solutions" to this are of the form $e^{\pm i\sqrt{\lambda}x}$ and linear combinations thereof. These can be called "generalized eigenfunctions", as they are not square-integrable. The generalized eigenfunction $v_w(x)=e^{2\pi iwx}$ has eigenvalue $\lambda=4\pi^2w^2\geq 0$. In some sense, all of $[0,\infty)$ are generalized eigenvalues. Thus, while we can certainly define eigenvalues and eigenfunctions in this way, they do not have the properties that we had in the Hilbert-Schmidt case, in which operators could be nicely decomposed using a discrete set of eigenvalues/functions.

Let $A\in\mathbb{R}^{n\times n}$ be symmetric, and with an orthonormal basis of eigenvectors $v_1,\dots,v_n$. For any $v\in\mathbb{R}^n$,

$$v=\sum_{k=1}^{n}v_kv_k^Tv,\qquad Av=\Big(\sum_{k=1}^{n}\lambda_kv_kv_k^T\Big)v$$

An analogy can be made with $\mathcal{L}$ self adjoint. Note that for $\mathcal{L}$ as the 1D Laplacian, this representation gives the Fourier inversion formula

$$f(x) = \int_{\mathbb{R}} e^{2\pi i w x} \hat{f}(w)\, dw, \qquad \nabla^2 f = \int_{\mathbb{R}} 4\pi^2 w^2 e^{2\pi i w x} \hat{f}(w)\, dw$$

Lecture 18: Nonlinear Approximation (4/14)

So far we have been mostly working with what may be called "linear" functional analysis. Polynomial interpolants are linear in the sense that the polynomial interpolant of a linear combination of functions is the linear combination of their individual polynomial interpolants. Kernel-based methods are linear in a sense. Differential and integral operators are linear as well.

Now, we consider the problem of nonlinear approximation. In particular, we will study

- Rational functions $f(x) \approx r(x) = \frac{a_0 + a_1 x + \ldots + a_m x^m}{b_0 + b_1 x + \ldots + b_n x^n}$. Such a function is called rational of type m, n.
- Composite functions built up from composing simple functions together $f(x) = (f_1 \circ \ldots \circ f_N)(x)$. Neural networks are an important example of these.

Rational Functions for Nonlinear Approximation

For $m \geq 0, n \geq 0$, let $\mathcal{R}_{m,n}$ be the set of rational functions of type m, n, meaning those of the form $r(x) = \sum_{k=0}^{m} a_k x^k / \sum_{j=0}^{n} b_j x^j$. Notably, $\mathcal{R}_{m,n}$ is not a vector space.

A rational function $r(x)$ is **degree** (or **exact type**) μ, v if

$$r(x) = \frac{\sum_{k=0}^{\mu} a_k x^k}{\sum_{j=0}^{v} b_j x^j}$$

where $a_\mu \neq 0$, $b_v = 1$, (i.e. numerator is degree μ, denominator is monic of degree v) and there are no common factors between the numerator and denominator. This definition does not define the degree of the zero function. For the most part, results will hold with the zero function having degree $\infty, 0$.

The above form with a division of arbitrary polynomials is somewhat difficult to work with. Instead, we will use the partial fraction form, which gives a somewhat "linear" structure that breaks the function into a sum of simple terms

$$r(x) = \sum_{k=1}^{v} \frac{c_k}{x - \xi_k} \qquad \xi_1, \ldots, \xi_k \text{ are distinct, } c_k \neq 0$$

this is given simply by Cauchy's integral formula. However, this formula requires finding zeros of the denominator and is numerically unstable. Thus, in practice, we tend to compute with the barycentric formula.

Let $E_{m,n} = \inf_{r_{m,n} \in \mathcal{R}_{m,n}} \|f - r_{m,n}\|_\infty$ be the error of the best rational approximation to f. Note that if $n = 0$, this is the error for the best polynomial interpolant.

Example 0.16. Let $f(x) = e^{-x^4}$ on $[-1,1]$. This is smooth, and we expect polynomials to do well. Numerically checking shows that indeed $E_{n,n}$ is about equal to $E_{2n,0}$ across n.

Let $f(x) = |x|$ on $[-1,1]$. This has a kink at $x = 0$, so polynomials do not do as well. Directly checking shows that $E_{2n,0} \sim 0.23/n$ while $E_{n,n} \sim 8e^{-\pi\sqrt{n}}$ is much lower.

Best Rational Approximation

Let $m, n \geq 0$ and $f : [-1,1] \to \mathbb{R}$ continuous. The best approximation is $r^* = \operatorname{argmin}_{r_{m,n} \in \mathcal{R}_{m,n}} \|f - r_{m,n}\|_\infty$. Recall that the error of the best polynomial interpolant of degree n equioscillates at least $n+2$ times. Likewise, $(f - r^*)(x)$ must equioscillate at least $m+n+2-d$ times.

The d is called the defect and is given by $d = \min(m-\mu, n-v)$, where μ, v is the degree of the best approximation. For instance, $r(x) = \frac{-x^3}{-1+x^2}$ is of exact type $(3,2)$. Then the defect is given by

$$d = \begin{cases} 0 & \text{in } \mathcal{R}_{3,2} \\ 0 & \text{in } \mathcal{R}_{3,3} \\ 1 & \text{in } \mathcal{R}_{4,3} \end{cases}$$

The equioscillation theorem for best rational approximations states that there exists a unique best rational approximation to a function f, and a rational function is the best approximation if and only if $f - r$ equioscillates at least $m+n+2-d$ times.

We will consider showing that equioscillation implies that r is the best. Suppose $f - r$ equioscillates at $x_0 < x_1 < \ldots < x_{m+n+1-d}$. For sake of contradiction suppose $\tilde{r} \in \mathcal{R}_{m,n}$ and $\|f - \tilde{r}\|_\infty < \|f - r\|_\infty$. Then $r - \tilde{r}$ takes nonzero values of alternating sign at the points $x_0, \ldots, x_{m+n+1-d}$. In particular, $r - \tilde{r}$ must have $m+n+1-d$ zeros. Write

$$r - \tilde{r} = \frac{p}{q} - \frac{\tilde{p}}{\tilde{q}} = \frac{p\tilde{q} - \tilde{p}q}{pq}$$

which has type $(\max(m+v, n+\mu), n+v)$ which simplifies to $(m+n-d, 2n-d)$. Since the numerator has at least $m+n+1-d$ zeros, it is in fact the zero function so $r - \tilde{r} = 0$ gives a contradiction.

The equioscillation result is very important. In polynomial approximation, due to linearity, we have results on polynomial approximation for classes of functions that form vector spaces, like C^∞. Due to nonlinearity, this cannot really be done for rational functions. Thus, we often have to consider rational approximations on a single function at a time. The equioscillation theorem is important for such analysis.

A Famous Problem

A famous problem comes from approximating $|x|$ on $[-1,1]$. Theorem 25.1 in ATAP states that

$$E_{n,0} \sim \frac{\beta}{n}, \quad \beta \approx 0.28$$

$$E_{n,n} \sim 8e^{-\pi\sqrt{n}}$$

This is useful since it can be used for other functions as well. Choose n even (so that $r_{n,n}$ has an even numerator and denominator, and there is no defect). Then we know that if $|x| \approx r_{n,n}(x)$ on

$[-1,1]$, then $|x| = r_{n/2,n/2}(x^2)$ on $[-1,1]$. In particular, $\sqrt{x^2} \approx r_{n/2,n/2}(x^2)$ on $[-1,1]$ implies that $\sqrt{x} \approx r_{n/2,n/2}(x)$ on $[0,1]$. This means that the approximation of $|x|$ and $\sqrt{x}$ are the same up to a change of variables. Also, this can be used for $\sqrt{1-x^2}$. Importantly, since $|x| = x\text{sign}(x)$, we can get results on functions that step. Writing $\text{ReLU}(x) = \max(0,x) = \frac{|x|+x}{2}$, we also have similarities with ReLU.

Recall that we went over the connection between quadrature and rational approximation earlier in the course. Note that $\frac{1}{|x|} = \int_0^\infty \frac{2}{\pi} \frac{dt}{t^2+x^2}$. Change variable $t = e^s$, so $dt = e^s ds$ and $|x| = \frac{2x^2}{\pi} \int_{-\infty}^{\infty} \frac{e^s ds}{e^{2s}+x^2}$. This has strong decay on the tails. It can be approximated by using a trapezoidal rule to fit a truncated version of this integral. This process gives a nice rational fit.

Lecture 19: Rational Approximation in Practice (4/16)

Although theoretically rational approximation gives more approximation strength than polynomial approximation, it can be difficult to compute rational approximations in practice. As in many other settings (e.g. neural networks), there are trade-offs to be taken between expressive power and ease of computation. In this lecture we go over methods to use rational approximation in practice.

Let $f : [-1,1] \to \mathbb{R}$ be a continuous function that we want to approximate. Given $m, n \geq 0$, suppose we have data $f(x_k)$, $0 \leq k \leq m+n$. We want a $p \in \mathbb{R}_m[x], q \in \mathbb{R}_n[x]$ such that $f(x_k) = p(x_k)/q(x_k)$ for $0 \leq k \leq m+n$. However, there are some issues with this problem that do not arise in polynomial approximation

- **Existence**: a rational interpolant of the desired order does not necessarily exist. For instance, there is no $r \in R_{1,1}$ such that $r(-1) = r(0) = 1$ and $r(1) = 2$. To see this, note that such an r is either a constant or a Mobius transform $r(x) = \frac{ax+b}{cx+d}$. As Mobious transforms are bijective, there is no such r.
- **Uniqueness**: p and q are obviously not unique, as they can be scaled, but this can be solved easily. However, there is another form of non-uniqueness: if $r \in R_{1,1}$ and $r(-1) = r(0) = r(1) = 0$, then setting $p = 0$ allows any q to be chosen.
- **Ill-posedness**: Say we want an $r \in R_{1,1}$ such that $r(-1) = 1+\epsilon$, $r(0) = 1$, and $r(1) = 1+2\epsilon$. When $\epsilon = 0$, take the constant function 1. If $\epsilon \neq 0$, then $r(x) = 1 + \frac{4/3\epsilon x}{x-1/3}$. Note that r always has a pole at $1/3$ is $\epsilon \neq 0$, but r has no poles when $\epsilon = 0$. In particular, if $\epsilon = 0$ but we have some rounding error on a computer so that we have a small nonzero ϵ, then a pole can be introduced.

Indeed, there is are spurious poles that arise in computing rational approximations in two main forms:

- Pole-zero pairs: in which a zero of q is almost a zero of p
- Pole with a small residual

To make rational interpolation more robust to these issues, we may:

- Linearize the problem: Instead of solving $f(x_k) = p(x_k)/q(x_k)$, solve $f(x_k)q(x_k) = p(x_k)$.

- Oversample
- Regularize

In linearizing the problem, the constraints form a linear system. Let $p(x) = \sum_{k=0}^{m} a_k T_k(x)$ and $q_x) = \sum_{k=0}^{n} b_k T_k(x)$ be written in the Chebyshev bases. The constraints can be written as

$$\begin{bmatrix} f(x_0) & & \\ & \ddots & \\ & & f(x_{m+n}) \end{bmatrix} \begin{bmatrix} q(x_0) \\ \vdots \\ q(x_{m+n}) \end{bmatrix} = \begin{bmatrix} p(x_0) \\ \vdots \\ p(x_{m+n}) \end{bmatrix}$$

$$\begin{bmatrix} f(x_0) & & \\ & \ddots & \\ & & f(x_{m+n}) \end{bmatrix} \begin{bmatrix} T_0(x_0) & \dots & T_n(x_0) \\ \vdots & \ddots & \vdots \\ T_0(x_{m+n}) & \dots & T_n(x_{m+n}) \end{bmatrix} \begin{bmatrix} b_0 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} p(x_0) \\ \vdots \\ p(x_{m+n}) \end{bmatrix}$$

$$= \begin{bmatrix} T_0(x_0) & \dots & T_{m+n}(x_0) \\ \vdots & \ddots & \vdots \\ T_0(x_{m+n}) & \dots & T_{m+n}(x_{m+n}) \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_m \\ 0 \end{bmatrix}$$

Inverting the Cheybshev matrix on the right, we have

$$\tilde{C} \begin{bmatrix} b_0 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} a_1 \\ \vdots \\ a_m \\ 0 \end{bmatrix}$$

This is an $m+n+1 \times n+1$ linear system. The last n equations do not depend on the a_j, so we can compute b by the equation

$$\tilde{C}[m+2 : \text{end}, :] \begin{bmatrix} b_0 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

this is an $n \times n+1$ system. We can use the SVD to find some b_i by least-squares. Also, the SVD can regularize, in the sense that thresholding

Connection to Neural Networks

Feedforward neural networks can be used as nonlinear function approximators. Consider a fully-connected feedforward neural network with $\sigma : \mathbb{R} \to \mathbb{R}$ a nonconstant, bounded, continuous activation function.

Theorem 25 (The Universal Approximation Theorem)**.** *Let $f : [0,1]^d \to \mathbb{R}$ be a continuous function and $\epsilon > 0$. There exists a one-layer neural network with σ activation, e.g. an integer $N \in \mathbb{N}$, real constants $u_i, b_i \in \mathbb{R}$ and $w_i \in \mathbb{R}^d$, such that the function $F(x) = \sum_{i=1}^{N} u_i \sigma(w_i^T x + b_i)$ satisfies $\big|F(x) - f(x)\big| < \epsilon$ for all $x \in [0,1]$*

However, N tends to be very large for any given ϵ. There are many empirical reasons to prefer deeper neural networks with more layers. In the next lectures, we will consider ReLU networks. Although the ReLU does not satisfy the conditions of the universal approximation theorem, one layer ReLU nets are still universal approximators. Note that rational functions are generally good for approximating smooth functions. On the other hand, ReLU networks are good at approximating piecewise linear functions. We will compute the number of neurons needed for ReLU networks to approximate certain smooth functions to some desired ϵ.

Lecture 20: Composition for Nonlinear Approximation (4/21)

Last time we discussed the universal approximation theorem. The proof, just as in some proofs of the Weierstrass theorem, is constructive, so a procedure to build a single-layer neural net is given by the prove. However, the number of nodes required for a given precision using the construction in the theorem is too large to be practical. This is analogous to the similar issue in the Weierstrass theorem—the degree of the approximating polynomial is not known.

Fix the activation to be $\sigma(x) = \text{ReLU}(x)$. We will use "neural nets" to refer to multi-layer perceptrons. Any ReLU net is a piecewise linear function. Note that in a single layer neural net, the number of pieces is on the order of the number of nodes. Adding more layers allows the number of pieces to scale multiplicatively in the product of the number of nodes in each layer.

Approximating the Quadratic

As a case-study, let us consider approximating $x \mapsto x^2$ on $[a, b]$ by a piecewise linear approximation. First, we consider the best linear approximant in the infinity norm, so the problem is of the form:

$$\min_{m,c} \left\| x^2 - mx - c \right\|_{\infty,[a,b]}$$

where $m, c \in \mathbb{R}$. This is simply the best linear polynomial approximant. Thus, the error equioscillates 3 times, say at points x_0, x_1, x_2, which gives a system

$$\begin{aligned} \delta &= x_0^2 - mx_0 - c \\ -\delta &= x_1^2 - mx_1 - c \\ \delta &= x_2^2 - mx_2 - c \end{aligned}$$

This is too many unknowns to solve for. However, we can show that $x_0 = a$ and $x_2 = b$. Then it can also be shown that x_1 has to be in the middle at $\frac{a+b}{2}$. This leaves us with three equations to solve for δ, m, and c. Solving this, we have that $|\delta| = \frac{(b-a)^2}{8}$.

Now, say we want a piecewise linear approximant $p(x)$ to x^2 on $[-1, 1]$. Let $\epsilon > 0$ be the precision that we want the error in the infinity norm to be bounded by. If $p(x)$ is piecewise linear with nodes $y_1 < \ldots < y_n$, then

$$\left\| x^2 - p(x) \right\|_\infty \geq \max_k \frac{(y_{k+1} - y_k)^2}{8}$$

Thus, the y_k must satisfy

$$\begin{aligned} \frac{(y_{k+1} - y_k)^2}{8} &\leq \epsilon \\ y_{k+1} - y_k &\leq \sqrt{8\epsilon} \end{aligned}$$

meaning the number of nodes N must satisfy $N \geq O(1/\sqrt{\epsilon})$. This makes one-layer ReLU nets look quite expensive, as even the best piecewise linear approximant requires a lot of nodes for the simple function x^2. However, ReLU nets can be saved by adding layers.

Theorem 26 (Yarotsky)**.** *The function $x \mapsto x^2$ on $[0,1]$ can be approximated to within $\epsilon > 0$ by a ReLU neural network with $O(\log(1/\epsilon))$ layers.*

Proof. Define the hat function $g : [0,1] \to [0,1]$ given by

$$g(x) = \begin{cases} 2x & x < 1/2 \\ 2(1-x) & x \geq 1/2 \end{cases}$$

Then define the composition $g_s(x) = (\underbrace{g \circ \ldots \circ g}_{s \text{ times}})(x)$, which is also piecewise linear. Let f_m be the piecewise linear interpolant to x^2 at $\frac{k}{2^m}, k = 0, \ldots, 2^m$, so $f_m(k/2^m) = k^2/2^{2m}$. Then the error can be computed

$$\left\| x^2 - f_m \right\|_\infty = 2^{-2m-2}$$

And the f_m satisfy the relationship

$$f_{m-1}(x) - f_m(x) = \frac{g_m(x)}{2^{2m}}$$

Hence, f_m satisfies

$$f_m(x) = x - \sum_{s=1}^{m} \frac{g_s(x)}{2^{2s}}$$

This is a neural net with m layers and g_s as an activation function. We can easily write $g(x)$ as a linear combination of ReLU activations,

$$g(x) = 2\text{ReLU}(x) - 4\text{ReLU}(x - 1/2) + 2\text{ReLU}(x - 1)$$

So $g(x)$ is a one-layer ReLU net. This means that g_s is an s-layer neural net. Thus, f_m is a ReLU neural network with $O(m)$ layers. Finally, to get the error $\left\| x^2 - f_m \right\|_\infty < \epsilon$, we need $2^{-2m-2} \leq \epsilon$, which gives

$$m \geq \frac{\log(1/\epsilon)}{2\log(2)} - 1 = O(\log(1/\epsilon))$$

□

Since $x \mapsto x^2$ can be approximated well, many other more interesting functions may be approximated well. Say we want to approximate $x \mapsto x^{1029}$. We can write $1029 = 10000000101$ in binary. Then $x^{1029} = x \cdot x^4 \cdot x^{1024}$, where x^4 and x^{1024} can be formed by 2 and 10 compositions of $x \mapsto x^2$, respectively.

Now, for a function $(x, y) \mapsto xy$, we can write $xy = \frac{1}{2}[(x+y)^2 - x^2 - y^2]$, which can easily be written as a neural network built from the x^2 nets. Thus, x^{1029} can be approximated efficiently by making networks for x^4 and x^{1024}, and then using the multiplicative xy net composition. In fact, this gives that any multivariate polynomial can then be approximated efficiently.

Lecture 21: Barron spaces and Hankel Operators (4/23)

Barron Spaces

Barron spaces are somewhat analogous to Sobolev spaces, but for neural networks. They were introduced by Barron in 1993 for old-school neural nets with sigmoid activations, but research in them has been reinvigorated by recent work. The (2019) paper "Barron Spaces and the Compositional Function Spaces for Neural Network Models" by Weinan E, Chao Ma, and Lei Wu works with these spaces.

Recall that a one-layer neural network takes the form $f(x) = \frac{1}{m}\sum_{j=1}^{m} a_j\sigma(b_j^T x + c_j)$, mapping $\mathbb{R}^d \to \mathbb{R}$, where the $\frac{1}{m}$ is placed there for convenience. The continuum analogue (letting $m \to \infty$, so the width $\to \infty$), is given by

$$f(x) = \int_{-\infty}^{\infty} a(s)\sigma(b(s)^T x + c(s))\, ds$$

where $a : \mathbb{R} \to \mathbb{R}$ and $b : \mathbb{R} \to \mathbb{R}^d$. A space is made of functions of this form, which are easier to analyze than the discrete version of one-layer neural nets with a bounded number of nodes.

Hankel Operators and AAK Theory

Let $(\alpha_j)_{j=0}^{\infty}$ be a sequence of complex numbers. We define

$$\Gamma_\alpha = \begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 & \dots \\ \alpha_1 & \alpha_2 & & & \\ \alpha_2 & & & & \\ \vdots & & & & \end{bmatrix}$$

For each Hankel operator Γ_α we may associate its corresponding Hankel function

$$k(z) = \frac{1}{z}\sum_{j=0}^{\infty} \alpha_j z^{-j}$$

Example 0.17.

$$\Gamma_\alpha = \begin{bmatrix} 1 & 1/2 & 1/4 \\ 1/2 & 1/4 & \\ 1/4 & & \end{bmatrix}$$

$k(z) = \frac{1}{z}\sum_{j=0}^{\infty} 2^{-j} z^{-j} = \frac{1}{z}\cdot\frac{1}{1-\frac{1}{2z}} = \frac{1}{z-\frac{1}{2}}$

$$\Gamma_\alpha = \begin{bmatrix} 1 & 1 & 1/2 & 1/4 \\ 1 & 1/2 & 1/4 & \\ 1/2 & 1/4 & & \\ 1/4 & & & \end{bmatrix}$$

$k(z) = \frac{1}{z} + \frac{1}{z^2}\sum_{j=0}^{\infty} 2^{-j} z^{-j} = \frac{1}{z} + \frac{1}{z(z-\frac{1}{2})} = \frac{z+\frac{1}{2}}{z(z-\frac{1}{2})}$

$$\Gamma_\alpha = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & \\ 1/3 & 1/4 & & \\ 1/4 & & & \end{bmatrix}$$

$k(z) = \sum_{j=0}^{\infty} \frac{1}{j+1} z^{-j}$.

We will use the following facts:

- AB and BA have the same nonzero eigenvalues

  Proof. For matrices $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{n\times m}$,

  $$\begin{bmatrix} I_m & \lambda A \\ 0 & I_n \end{bmatrix}^{-1} \begin{bmatrix} I_m - \lambda AB & 0 \\ -B & I_n \end{bmatrix} \begin{bmatrix} I_m & \lambda A \\ 0 & I_n \end{bmatrix} = \begin{bmatrix} I_m & 0 \\ -B & I_n - \lambda BA \end{bmatrix}$$

  From this similarity relation, one can show that show the claim easily. □

- Let $A \in \mathbb{C}^{n\times n}$. Then, (ignoring convegence issues)

  $$(zI - A)^{-1} = \sum_{j=1}^{\infty} A^{j-1} z^{-j}$$

  Proof.

  $$\begin{aligned}(zI - A)\sum_{j=1}^{\infty} A^{j-1} z^{-j} &= \sum_{j=1}^{\infty} A^{j-1} z^{-j+1} - \sum_{j=1}^{\infty} A^j z^{-j} \\ &= I\end{aligned}$$

  □

- Let $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times k}$, and $X \in \mathbb{R}^{n\times n}$ is unknown. The solution to

  $$X - AXA^* = BB^*$$

  is given by $X = \sum_{k=0}^{\infty} A^k BB^* (A^*)^k$.

  Proof. Substitute this X into the equation and see that it holds. □

Now, assume that we are considering a Hankel operator where $k(z)$ is a rational function. Note that this includes the first and second examples above, but not the Hilbert operator. Write $k(z)$ as

$$k(z) = \sum_{j=1}^{n} \frac{b_j}{z - a_j}$$

with distinct poles, and $b_j \neq 0$. We may write this as

$$\begin{aligned} k(z) &= C(zI - A)^{-1}B \\ A &= \begin{bmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{bmatrix} \\ B &= \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \\ C &= \begin{bmatrix} 1 & \dots & 1 \end{bmatrix} \end{aligned}$$

Theorem 27 (Kronecker)**.** *The rank of a Hankel operator is equal to the number of poles of the corresponding Hankel function.*

Proof. We use the above linear algebra facts to prove this.

$$\begin{aligned} k(z) &= C(zI - A)^{-1}B \\ &= \sum_{j=1}^{\infty} CA^{j-1}Bz^{-j} \\ &= \frac{1}{z}\sum_{j=1}^{\infty} \alpha_j z^{-j} \end{aligned}$$

so that $\alpha_j = CA^jB$. Now, note that $(H_\alpha)_{j,k} = \alpha_{j+k} = CA^{j+k}B = (CA^j)(A^kB)$. The left matrix only depends on the row index, and the right only on the column. Thus, we have

$$H_\alpha = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix} \begin{bmatrix} B & AB & A^2B & \dots \end{bmatrix}$$

The left factor is $\infty \times n$, and the right factors is $n \times \infty$. This means that $\operatorname{rank}(H_\alpha) \leq n$. This argument can be reversed to get the other inequality. □

Lectures 22-24: Stochastic Optimization

`https://gowerrobert.github.io/pdf/teaching/cornell/lectures_cornell.pdf`

Lecture 25: AAK Theory

We return to studying Hankel operators and the corresponding Hankel functions. Assume that $k(z)$ is a rational function with all of its poles in the unit disk. Then we have a special case of the AAK theorem:

Theorem 28 (AAK)**.** *Let $k(z) = C(zI - A)^{-1}B$ have n poles. Then*

$$\sigma_k(\Gamma_\alpha)^2 = \lambda_k(PQ) \qquad 1 \le k \le n$$

where PQ is $n \times n$. These matrices take the form

$$P = \Xi\Xi^* \quad \Xi = \begin{bmatrix} B & AB & A^2B & \dots \end{bmatrix}$$
$$Q = \Omega^*\Omega \quad \Omega = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix}$$

Note that $\sigma_k(\Gamma_\alpha) = 0$ for $k > n$ by Kronecker's theorem. This gives a case in which we may compute all of the singular values of an infinite operator by finding the eigenvalues of a finite matrix. Of course, we cannot compute the P and Q naively, as they are given as the product of infinite matrices.

Proof. As in the proof of Kronecker's theorem,

$$\begin{aligned} k(z) &= C(zI - A)^{-1}B \\ &= \sum_{j=1}^{\infty} CA^{j-1}Bz^{-j} \\ (\Gamma_\alpha)_{jk} &= \alpha_{j+k} \\ &= CA^{j+k}B \\ &= (CA^j)(A^kB) \end{aligned}$$

Thus, we have that

$$\Gamma_\alpha = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \end{bmatrix} \begin{bmatrix} B & AB & A^2B & \dots \end{bmatrix} = \Omega\Xi$$

Let $1 \le k \le n$. Then

$$\begin{aligned} \sigma_k(\Gamma_\alpha)^2 &= \sigma_k(\Omega\Xi)^2 \\ &= \lambda_k(\Xi^*\Omega^*\Omega\Xi) \\ &= \lambda_k(\Xi\Xi^*\Omega^*\Omega) \\ &= \lambda_k(PQ) \end{aligned}$$

where we use tha that nonzero eigenvalues of AB are the same as those of BA. □

To actually compute P, we note that

$$P = \sum_{j=0}^{\infty} A^j BB^*(A^j)^*$$

Thus, P is the solution to $P - APA^* = BB^*$. Similarly,

$$Q = \sum_{j=0}^{\infty} (A^j)^* C^* C A^j$$

so Q is the solution to $Q - A^*QA = C^*C$. To actually solve this equation, consider the equation for P, and note that the left hand side is linear in P. We can write out the equation as a big matrix equation. If A is diagonal, this is simple, since then the entries of P satisfy

$$p_{jk} - a_j p_{jk} \overline{a}_k = b_j \overline{b}_k$$

and thus they can be solved independently

$$p_{jk} = \frac{b_j \overline{b}_k}{1 - a_j \overline{a}_k}$$

Hence, we have an algorithm for determining the singular values of Γ_α.

1. Suppose we have $k(z) = C(zI - A)^{-1}B$ with poles in the unit disk.
2. Solve $X - AXA^* = BB^*$ for P.
3. Solve $X - A^*XA = C^*C$ for Q.
4. Compute the eigenvalues of PQ.

This is an $O(n^3)$ algorithm for computing the singular values of an infinite matrix with rank n, so it has asymptotically the same cost as that of computing the singular values of an $n \times n$ matrix.

AAK theory connects singular values (restricted rank approximation) with rational approximation. Let $R_k(D)$ be the set of rational functions such that the number of poles of r in D (the unit disk) is at most k. Then it holds that

$$\sigma_{k+1}(\Gamma_\alpha) = \inf_{r \in R_k(D)} \|k - r\|_{\infty, D} = \inf_{r \in R_k(D)} \sup_{\theta \in [0, 2\pi)} \left| k(e^{i\theta}) - r(e^{i\theta}) \right|$$

this means that the best rank k approximation problem of a Hankel operator is related to best rational approximation of a Hankel function on the unit disk. Also, the best rank k approximation to an infinite Hankel operator is in fact Hankel. However, this does not hold for finite Hankel matrices. This is really interesting, as this means the SVD preserves Hankel structure, which does not usually hold for structured matrices.