Week 8: Friday, Oct 12

Why eigenvalues?

I spend a lot of time thinking about eigenvalue problems. In part, this is because I look for problems that can be solved via eigenvalues. But I might have fewer things to keep me out of trouble if there weren't so many places where eigenvalue analysis is useful! The purpose of this lecture is to tell you about a few applications of eigenvalue analysis, or perhaps to remind you of some applications that you've seen in the past.

1 Nonlinear equation solving

The eigenvalues of a matrix are the roots of the characteristic polynomial

$$p(z) = \det(zI - A).$$

One way to compute eigenvalues, then, is to form the characteristic polynomial and run a root-finding routine on it. In practice, this is a terrible idea, if only because the root-finding problem is often far more sensitive than the original eigenvalue problem. But even if sensitivity were not an issue, finding *all* the roots of a polynomial seems like a nontrivial undertaking. Iterations like Newton's method, for example, only converge locally. In fact, the `roots` command in MATLAB computes the roots of a polynomial by finding the eigenvalues of a corresponding *companion matrix* with the polynomial coefficients on the first row, ones on the first subdiagonal, and zeros elsewhere:

$$C = \begin{bmatrix} c_{d-1} & c_{d-2} & \dots & c_1 & c_0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix}.$$

The characteristic polynomial for this matrix is precisely

$$\det(zI - C) = z^d + c_{d-1}z^{d-1} + \ldots + c_1 z + c_0.$$

There are some problems that connect to polynomial root finding, and thus to eigenvalue problems, in surprising ways. For example, the problem of

finding "optimal" rules for computing integrals numerically (sometimes called Gaussian quadrature rules) boils down to finding the roots of orthogonal polynomials, which can in turn be converted into an eigenvalue problem; see, for example, "Calculation of Gauss Quadrature Rules" by Golub and Welsch (*Mathematics of Computation*, vol 23, 1969).

More generally, eigenvalue problems are one of the few examples I have of a nonlinear equation where I can find *all* solutions in polynomial time! Thus, if I have a hard nonlinear equation to solve, it is very tempting to try to massage it into an eigenvalue problem, or to approximate it by an eigenvalue problem.

2 Optimization

Recall that the matrix 2-norm is defined as

$$\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|.$$

Taking squares and using the monotonicity of the map $z \to z^2$ for non-negative arguments, we have

$$\|A\|_2^2 = \max_{\|x\|^2=1} \|Ax\|^2 = \max_{x^T x = 1} x^T A^T A x.$$

The x that solves this constrained optimization problem must be a stationary point for the augmented Lagrangian function

$$L(x, \lambda) = x^T A^T A x - \lambda(x^T x - 1),$$

i.e.

$$\begin{aligned} \nabla_x L(x, \lambda) &= 2(A^T A x - \lambda x) = 0 \\ \nabla_\lambda L(x, \lambda) &= x^T x - 1 = 0. \end{aligned}$$

These equations say that x is an eigenvector of $A^T A$ with eigenvalue λ. The largest eigenvalue of $A^T A$ is therefore $\|A\|_2^2$.

More generally, if H is any Hermitian matrix, the *Rayleigh quotient*

$$\rho_H(v) = \frac{v^* H v}{v^* v}$$

has stationary points exactly when v is an eigenvector of H. Optimizing the Rayleigh quotient is therefore example of a *non-convex* global optimization problem that I know how to solve in polynomial time. Such examples are rare, and so it is tempting to try to massage other nonconvex optimization problems so that they look like Rayleigh quotient optimization, too.

To give an example of a nonconvex optimization that can be usefully approximated using Rayleigh quotients, consider the superficially unrelated problem of graph bisection. Given an undirected graph G with vertices V and edges $E \subset V \times V$, we want to find a partition of the nodes into two equal-size sets such that few edges go between the sets. That is, we want to write V as a disjoint union $V = V_1 \cup V_2$, $|V_1| = |V_2|$, such that the number of edges cut $|E \cap (V_1 \times V_2)|$ is minimized. Another way to write the same thing is to label each node i in the graph with $x_i \in \{+1, -1\}$, and define V_1 to be all the nodes with label $+1$, V_2 to be all the nodes with label -1. Then the condition that the two sets are the same size is equivalent to

$$\sum_i x_i = 0,$$

and the number of edges cut is

$$\frac{1}{4} \sum_{(i,j) \in E} (x_i - x_j)^2$$

We can rewrite the constraint more concisely as $e^T x = 0$, where e is the vector of all ones; as for the number of edges cut, this is

$$\text{edges cut} = \frac{1}{4} x^T L x$$

where the *graph Laplacian* L has the node degrees on the diagonal and -1 in off-diagonal entry (i, j) iff there is an edge from i to j.

Unsurprisingly, the binary quadratic programming problem

$$\text{minimize } x^T L x \text{ s.t. } e^T x = 0 \text{ and } x \in \{+1, -1\}^n$$

is NP-hard, and we know of no efficient algorithms that are guaranteed to work for this problem in general. On the other hand, we can *relax* the problem to

$$\text{minimize } v^T L v \text{ s.t. } e^T v = 0 \text{ and } \|v\|^2 = n, v \in \mathbb{R}^n,$$

and this problem is an eigenvalue problem: v is the eigenvector associated with the smallest positive eigenvalue of L, and $v^T L v$ is n times the corresponding eigenvalue. Since the constraint in the first problem is strictly stronger than the constraint in the second problem, $n\lambda_2(L)$ is in fact a lower bound on the smallest possible cut size, and the sign pattern of v often provides a partition with a small cut size. This is the heart of *spectral partitioning* methods.

3 Dynamics

Eigenvalue problems come naturally out of separation of variables methods, and out of transform methods for the dynamics of discrete or continuous linear time invariant systems, including examples from physics and from probability theory. They allow us to analyze complicated high-dimensional dynamics in terms of simpler, low-dimensional systems. We consider two examples: separation of variables for a free vibration problem, and convergence of a discrete-time Markov chain.

3.1 Generalized eigenvalue problems and free vibrations

One of the standard methods for solving differential equations is *separation of variables*. In this approach, we try to write special solutions as a product of simpler functions, and then write the equations that those functions have to satisfy. As an example, consider a differential equation that describes the free vibrations of a mechanical system:

$$M\ddot{u} + Ku = 0$$

Here $M \in \mathbb{R}^{n\times n}$ is a symmetric positive definite *mass matrix* and $K \in \mathbb{R}^{n\times n}$ is a symmetric *stiffness matrix* (also usually positive definite, but not always). We look for solutions to this system of the form

$$u(t) = u_0 \cos(\omega t),$$

where u_0 is a fixed vector. To have a solution of this form, we must have

$$Ku_0 - \omega^2 M u_0 = 0,$$

i.e. (ω^2, u_0) is an eigenpair for a *generalized* eigenvalue problem. In fact, the eigenvectors for this generalized eigenvalue problem form an M-orthonormal basis for $\mathbb{R}^n$, and so we can write *every* free vibration as a linear combination of these simple "modal" solutions.

3.2 Markov chain convergence and the spectral gap

This high-level idea of using the eigenvalue decomposition to understand dynamics is not limited to differential equations, nor to mechanical systems. For example, a *discrete-time Markov chain* on n states is a random process where the state X_{k+1} is a random variable that depends only on the state X_k. The *transition matrix* for the Markov chain is a matrix P where P_{ij} is the (fixed) probability of transitioning to state i from state j, i.e.

$$P_{ij} = P\{X_{k+1} = j | X_k = i\}.$$

Let $\pi^{(k)} \in \mathbb{R}^n$ be the distribution vector at time k, i.e.

$$\pi_i^{(k)} = P\{X_k = i\}.$$

Then we have the recurrence relationship

$$(\pi^{(k+1)})^T = (\pi^{(k)})^T P.$$

In general, this means that

$$(\pi^{(k)})^T = (\pi^{(0)})^T P^k.$$

Now, suppose the transition matrix P is diagonalizable, i.e. $P = V\Lambda V^{-1}$. Then

$$P^k = V\Lambda V^{-1} V\Lambda V^{-1} \ldots V\Lambda V^{-1} = V\Lambda \ldots \Lambda V^{-1} = V\Lambda^k V^{-1},$$

and so

$$(\pi^{(k)})^T = (\pi^{(0)})^T V\Lambda^k V^{-1}.$$

An *ergodic* Markov chain has one eigenvalue at one, and all the other eigenvalues are less than one in modulus. In this case, the row eigenvector associated with the eigenvalue at one can be normalized so that the coefficients are all positive and sum to 1. This normalized row eigenvector $\pi^{(*)}$ represents the

stationary distribution to which the Markov chain eventually converges. To compute the rate of convergence, one looks at

$$\|(\pi^{(k)} - \pi^{(*)})^T\| = \|(\pi^{(0)} - \pi^{(*)})^T (V\tilde{\Lambda}^k V^{-1})\| \leq \|(\pi^{(0)} - \pi^{(*)})^T\| \, \kappa(V) \|\tilde{\Lambda}\|^k$$

where $\Lambda = \mathrm{diag}(1, \lambda_2, \lambda_3, \ldots)$, $|\lambda_i| \geq |\lambda_{i+1}|$, and $\tilde{\Lambda} = \mathrm{diag}(0, \lambda_2, \lambda_3, \ldots)$. In most reasonable operator norms, $|\tilde{\Lambda}|^k = |\lambda_2|^k$, and so a great deal of the literature on convergence of Markov chains focuses on $1 - |\lambda_2|$, called the *spectral gap*. But note that this bound does not depend on the eigenvalues alone! The condition number of the eigenvector matrix also plays a role, and if $\kappa(V)$ is very large, then it may take a long time indeed before anyone sees the asymptotic behavior reflected by the spectral gap.

4 Deductions from eigenvalue distributions

In most of our examples so far, we have considered both the eigenvalues and the eigenvectors. Now let us turn to a simple example where the distribution of eigenvalues can be illuminating.

Let A be the adjacency matrix for a graph, i.e.

$$A_{ij} = \begin{cases} 1, & \text{if there is an edge from } i \text{ to } j \\ 0, & \text{otherwise.} \end{cases}$$

Then $(A^k)_{ij}$ is the number of paths of length k from node i to node j. In particular, $(A^k)_{ii}$ is the number of cycles of length k that start and end at node i, and $\mathrm{trace}(A^k)$ is the total number of length k cycles starting from any node. Recalling that the trace of a matrix is the sum of the eigenvalues, and that the eigenvalues of a matrix power are the power of the eigenvalues, we have that

$$\#\text{ paths of length } k = \sum_i \lambda_i(A)^k,$$

where $\lambda_i(A)$ are the eigenvalues of A; and asymptotically, the number of cycles of length k for very large k scales like $\lambda_1(A)^k$, where $\lambda_1(A)$ is the largest eigenvalue of the matrix A.

While the statement above deals only with eigenvalues and not with eigenvectors, we can actually say more if we include the eigenvector; namely, if the graph A is irreducible (i.e. there is a path from every state to every other state), then the largest eigenvalue $\lambda_1(A)$ is a real, simple eigenvalue,

and asymptotically the number of paths from any node i to node j scales like the (i, j) entry of the rank one matrix

$$\lambda_1^k v w^T$$

where v and w are the column and row eigenvectors of A corresponding to the eigenvalue λ_1, scaled so that $w^T v = 1$.