10
Markov Chain Mixing, Expanders & Eigenvalues

10.1 Introduction

In the last lecture, we looked at Markov Chain Monte Carlo (MCMC) methods. The goal of MCMC is often to sample from the stationary distribution π^* of a Markov Chain (MC).

We know that if a Markov Chain M is irreducible and aperiodic (ergodic), then for any starting distribution $\pi^{(0)}$, the distribution converges to the stationary distribution:

$$\pi^{(t)} \to \pi^* \quad \text{as } t \to \infty.$$

The crucial question for algorithmic applications is: **How fast does the chain "mix"?**

Definition 10.1 (Mixing Time)**.** The mixing time is the time t required such that the total variation distance between the current distribution and the stationary distribution is small:

$$\|\pi^{(t)} - \pi^*\|_{TV} \leq \epsilon.$$

We previously saw that the *coupling time* provides an upper bound on the mixing time. (We mentioned the analysis of Glauber dynamics for graph colorings, showing that with $k \geq 4\Delta + 1$ colors, the mixing time is $O(kn \log(n/\epsilon))$).

In this lecture, we explore another powerful technique for analyzing mixing time based on the eigenvalues of the transition matrix.

- This leads to the concept of **Expander Graphs**—graphs which have a good "eigenvalue gap".
- We will explore another view of expanders (combinatorial).
- We will see an application of expanders for probability amplification and error correcting codes.

10.2 Mixing Time and Eigenvalues

Let's recap the evolution of a Markov Chain with transition matrix P. The distribution at time t is $\pi^{(t)} = \pi^{(0)}P^t$. The stationary distribution satisfies $\pi^* = \pi^* P$ (it is a left eigenvector with eigenvalue 1).

We want to understand the convergence rate of $\pi^{(t)}$ to π^*.

$$\begin{aligned}\pi^{(t)} - \pi^* &= \pi^{(0)}P^t - \pi^* P^t \quad (\text{Since } \pi^* = \pi^* P^t) \\ &= (\pi^{(0)} - \pi^*)P^t.\end{aligned}$$

The behavior of P^t determines how quickly the initial difference $(\pi^{(0)} - \pi^*)$ decreases. This depends on the eigenvalue structure of P.

10.2.1 Reversible Markov Chains and the Spectrum

We focus on an important class of MCs that simplifies the spectral analysis.

Definition 10.2 (Time Reversible Markov Chains)**.** A Markov Chain is (time) reversible if there exists a distribution π such that the detailed balance equations hold for all states i, j:

$$\pi_i P_{ij} = \pi_j P_{ji}.$$

If this holds, π is the stationary distribution.

A subclass is **Symmetric Chains**, where $P_{ij} = P_{ji}$. These are clearly reversible, with the uniform distribution $\pi_i = 1/n$ serving as the certificate.

If a chain is symmetric, P is a symmetric matrix, so all its eigenvalues are real.

If the chain is reversible, P is similar to a symmetric matrix. Specifically, if $D = \text{diag}(\pi)$, then $D^{1/2}PD^{-1/2}$ is symmetric. Since similar matrices share the same eigenvalues, the eigenvalues of P are also real.

We can now state the properties of the spectrum (the set of eigenvalues) for these chains, using results related to the Perron-Frobenius theorem.

Theorem 10.3. *Let P be the transition matrix of an irreducible, aperiodic, and reversible Markov Chain. Then the eigenvalues satisfy:*

$$1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq -1.$$

Furthermore:

1. *All eigenvalues are real (by reversibility).*
2. *$\lambda_1 = 1$ is unique (by irreducibility). The corresponding eigenvector is related to π^*. This implies $\lambda_2 < 1$.*
3. *$\lambda_n > -1$ (by aperiodicity).*

10.2.2 *Convergence Rate and the Spectral Gap*

If $|\lambda_i| < 1$ for all $i > 1$, then $\pi^{(t)} \to \pi^*$ as $t \to \infty$. The rate of convergence depends on the second largest eigenvalue in magnitude.

Definition 10.4 (Spectral Gap)**.** The second largest eigenvalue magnitude is:

$$\lambda_* = \max_{i \geq 2} |\lambda_i| = \max(|\lambda_2|, |\lambda_n|).$$

The **spectral gap** is $1 - \lambda_*$.

Proof Sketch of Convergence (Simplified case: P is symmetric). If P is symmetric, $\pi^* = \text{uniform} = \frac{1}{n}\mathbf{1}$. P has an orthonormal eigenbasis $v_1, v_2, \ldots, v_n$. We can choose v_1 corresponding to $\lambda_1 = 1$ (e.g., $v_1 = \frac{1}{\sqrt{n}}\mathbf{1}$). The other eigenvectors $v_2, \ldots, v_n$ are orthogonal to v_1.

We analyze the difference vector $\pi^{(0)} - \pi^*$. Since $\sum_j \pi_j^{(0)} = 1$ and $\sum_j \pi_j^* = 1$, the sum of the components of the difference vector is 0. This means the difference vector is orthogonal to v_1.

$$\pi^{(0)} - \pi^* = \sum_{i \geq 2} c_i v_i.$$

Now we apply P^t:

$$(\pi^{(0)} - \pi^*)P^t = \sum_{i \geq 2} c_i v_i P^t = \sum_{i \geq 2} c_i \lambda_i^t v_i.$$

We look at the L2 norm:

$$\|(\pi^{(0)} - \pi^*)P^t\|_2^2 = \sum_{i \geq 2} c_i^2 \lambda_i^{2t} \leq \lambda_*^{2t} \sum_{i \geq 2} c_i^2 = \lambda_*^{2t} \|\pi^{(0)} - \pi^*\|_2^2.$$

Since $\lambda_* < 1$, this goes to 0 as $t \to \infty$. □

This analysis can be formalized to bound the mixing time in the Total Variation distance.

Theorem 10.5 (Mixing Time Bound)**.** *Suppose we start at state x. For a reversible, irreducible, aperiodic MC:*

$$\|\pi^{(t)} - \pi^*\|_{TV} \leq O\left(\frac{\lambda_*^t}{\sqrt{\pi^*(x)}}\right).$$

To get the TV distance down to ϵ, it suffices to take:

$$t = O\left(\frac{\log(1/\pi^*(x)) + \log(1/\epsilon)}{1 - \lambda_*}\right)$$

steps. The mixing time is inversely proportional to the spectral gap. We want the gap to be large.

Lazy Chains. Often, we analyze a "lazy" version of the chain. With probability $1/2$, it stays at the current state; with probability $1/2$, it makes a move according to P. The new transition matrix is $P' = \frac{1}{2}(I + P)$. This shifts the eigenvalues upwards, ensuring all eigenvalues are non-negative ($\lambda_n \geq 0$). In this case, $\lambda_* = \lambda_2$.

10.3 Expander Graphs

A Markov Chain mixes fast if it has a large spectral gap. A special case of MCs is random walks on graphs.

Consider a random walk on a d-regular graph $G = (V, E)$. The transition matrix is related to the adjacency matrix A. For a lazy walk, $P = \frac{1}{2}I + \frac{1}{2d}A$.

Definition 10.6 (Spectral Expander)**.** A graph where the transition matrix P has a large spectral gap (say, constant) is called a **spectral expander**.

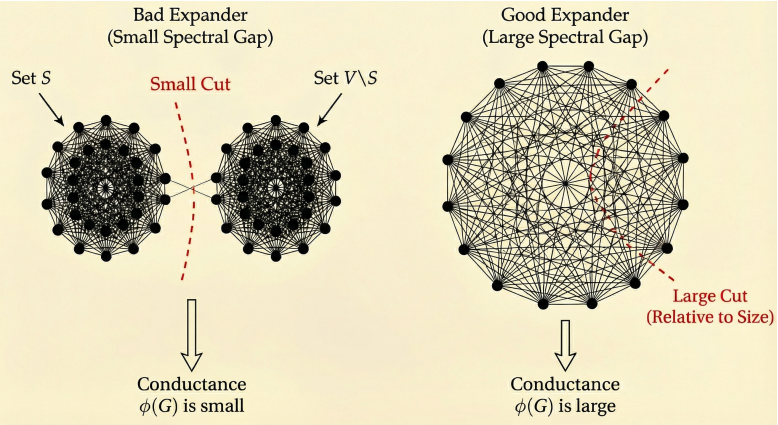
There is another, combinatorial notion of expansion.

10.3.1 Combinatorial Expansion

Definition 10.7 (Vertex Expansion)**.** A graph is a vertex expander if every subset of vertices S that is not too large (e.g., $|S| \leq n/2$) "expands" significantly.

$$|N(S) \setminus S| \geq \alpha |S|$$

for some constant $\alpha > 0$. $N(S)$ is the set of neighbors of S.



Definition 10.8 (Edge Expansion / Conductance)**.** The edge expansion (or conductance) $\phi(G)$ measures how well connected the

(d-regular) graph is. It is defined as:

$$\phi(G) = \min_{S \subseteq V} \frac{|E(S, \bar{S})|}{\min(|S|, |\bar{S}|)}.$$

10.3.2 Spectral vs. Combinatorial Expansion

There is a fundamental connection between these two views:

Spectral expansion $\leftrightsquigarrow$ combinatorial expansion.

(While the notions are qualitatively the same, one must be careful with the quantitative bounds). This connection was explored by Noga Alon and Jeff Cheeger (search Cheeger's inequalities) and there has been a lot of generalizations since then.

Why do we care about this equivalence? It provides two different sets of techniques (linear algebraic and combinatorial) to prove expansion and analyze mixing.

10.3.3 Why Care About Expanders?

Expanders behave like "random-like" graphs, despite potentially having low degree. They have numerous applications:

- Randomness amplification.
- Expander decompositions (used in algorithms).
- Expander codes (error correction).

10.3.4 Constructions of Expanders

Do graphs with constant spectral gap and constant degree exist? Yes.

- **Probabilistic Constructions:**
  - A random constant-degree graph is an expander with high probability (w.h.p.).
  - Taking the union of 3 random perfect matchings results in an expander w.h.p.
- **Explicit Constructions (Algebraic/Number Theoretic):**
  - Cayley graphs: e.g., consider $\mathbb{Z}_p$ and connect x to $x+1, x-1$, and $1/x$.
  - Gabber-Galil Expander (based on work by Margulis): Based on maps on a grid/torus, e.g., $(x, y) \mapsto (x, x \pm y)$ and $(x, y) \mapsto (x \pm y, y)$.

  - Ramanujan graphs (Lubotzky, Phillips, Sarnak): These are "optimal" expanders.

- **Combinatorial Constructions (CS):**
  - Zig-zag product (Reingold, Vadhan, Wigderson).
  - Random Lifts (Bilu, Linial).

10.4 Application: Randomness Amplification

Suppose we have a randomized algorithm A that uses r random bits. On any input X, it is correct with high probability:

$$\Pr(A \text{ is wrong on input } X) \leq \delta \quad (\text{e.g., } \delta = 1/100).$$

We want to reduce the error probability to δ^k.

The Standard Way (Independent Repetitions). Run the algorithm $T = O(k)$ times using T independent r-bit strings. Take the majority vote. By Chernoff bounds, the probability that the majority is incorrect is $\leq \delta^k$. **Total randomness used:** $O(rk)$ bits.

The Expander Way (Reduced Randomness). We can achieve the same error reduction using much less randomness.

Let G be a constant-degree (Δ) expander graph on $N = 2^r$ nodes.

Algorithm 5: Randomness Amplification via Expander Walk

5.1 1. Pick an initial random string x_0 of length r. View this as a node in G. (Uses r bits).

5.2 2. Perform a random walk of length T starting at x_0: $x_0, x_1, \ldots, x_T$. (Uses $O(T \log \Delta) = O(T)$ bits).

5.3 3. Run the algorithm using these strings: $A(x_0), A(x_1), \ldots, A(x_T)$.

5.4 4. Return the majority vote.

Total randomness used: Only $r + O(T) = r + O(k)$ bits!

10.4.1 Analysis

Theorem 10.9. *If G is a good enough expander (e.g., $|\lambda_*| \leq 1/10$, assuming $\delta = 1/100$), then the probability of error is still small (e.g., $\leq \delta^k$ if $T = O(k)$).*

Proof Sketch. If the majority vote is incorrect, the algorithm must have made a mistake on $\geq T/2$ of the strings x_i.

Let P be the transition matrix of the random walk on the expander. Let $B \subset V$ be the set of "bad" strings for which the algorithm fails. We know $|B|/N \leq \delta$.

Define Z as a diagonal matrix where $Z_{ii} = 1$ if $i \in B$ (a bad string), and $Z_{ii} = 0$ otherwise.

The analysis relies on the fact that random walks on expanders rapidly approach the uniform distribution, reducing the correlation between the samples x_i.

Lemma 10.10 (Contraction Lemma)**.** *Let π be any probability distribution. For the parameters assumed ($|\lambda_*| \leq 1/10, \delta = 1/100$), we have:*

$$\|\pi PZ\|_2 \leq c\|\pi\|_2,$$

where $c < 1$ (e.g., $c = 1/5$).

Proof Sketch of Lemma. Let π^* be the uniform distribution (the stationary distribution). We decompose $\pi = x + y$, where x is the component along π^* and y is orthogonal to π^*.

$$\|\pi PZ\|_2 = \|(x+y)PZ\|_2 \leq \|xPZ\|_2 + \|yPZ\|_2.$$

- Term (a): $\|xPZ\|_2 = \|xZ\|_2$ (since $xP = x$). This term is small because x is uniform and Z represents the small bad set B. It is bounded by $\sqrt{\delta}\|x\|_2 = (1/10)\|x\|_2$.
- Term (b): $\|yPZ\|_2 \leq \|yP\|_2$. Since y is orthogonal to the principal eigenvector, P contracts it by the spectral gap: $\|yP\|_2 \leq \lambda_*\|y\|_2 = (1/10)\|y\|_2$.

Combining these, $\|\pi PZ\|_2 \leq \frac{1}{10}(\|x\|_2 + \|y\|_2)$. Using Cauchy-Schwarz, this is $\leq \frac{\sqrt{2}}{10}\|\pi\|_2 < \frac{1}{5}\|\pi\|_2$. □

Now we analyze the probability of being incorrect at $t = T/2$ specific steps during the walk. Assume we start from the uniform distribution π^0. The probability of failing at a sequence of steps corresponding to the walk where t specific steps are bad is bounded by the norm of the product of matrices (involving t occurrences of PZ):

$$\|\pi^0 \dots PZ \dots PZ \dots\|_1$$

We bound the L_1 norm using the L_2 norm: $\leq \sqrt{N}\|\dots\|_2$. Applying the Contraction Lemma repeatedly (for the t occurrences of PZ combinations):

$$\leq \sqrt{N} \cdot \left(\frac{1}{5}\right)^t \cdot \|\pi^0\|_2.$$

Since π^0 is uniform, $\|\pi^0\|_2 = 1/\sqrt{N}$. The probability is $\leq (1/5)^t = (1/5)^{T/2}$.

Finally, we take a union bound over all $\binom{T}{T/2} \approx 2^T$ choices of $T/2$ steps where failure occurs:

$$\Pr(\text{wrong on } \geq T/2 \text{ steps}) \leq 2^T \left(\frac{1}{5}\right)^{T/2} = \left(\frac{2}{\sqrt{5}}\right)^T.$$

Since $2/\sqrt{5} < 1$, this is exponentially small in T. By setting $T = O(k)$, we can achieve the desired error bound δ^k. □

Remark 10.11. If λ_* is not small enough, we can instead walk α steps between samples. We choose α such that $(\lambda_*)^\alpha$ is small enough. This only increases the cost by a constant factor.

10.5 Application: Expander Codes

Another application of expanders is in the construction of Error Correcting Codes (ECCs), initiated by Gallager and Tanner, and later developed by Sipser and Spielman, followed by many other works.

Goal: Find a codeword set $C \subseteq \{0,1\}^n$ such that:

1. C is large (Rate is $\Omega(1)$).
2. The Hamming distance between codewords is large (Distance is $\Omega(n)$).
3. Efficient encoding and decoding algorithms exist.

We use bipartite expanders to construct good codes (Tanner Codes).

10.5.1 Bipartite Expanders

Consider a bipartite graph $G = (L \cup R, E)$ such that $|L| = n$ and $|R| = m$. Assume it is left-regular with degree d.

Definition 10.12 ((α,β)-Expander)**.** G is an (α,β)-expander if for every subset $S \subseteq L$ such that $|S| \leq \alpha n$, the neighborhood expands by a factor β:

$$|N(S)| \geq \beta |S|.$$

Fact 10.13. There exist constructions of constant-degree bipartite graphs where α is a constant, $m = O(n)$, and the expansion factor is close to the maximum possible, e.g., $\beta \geq \frac{3d}{4}$.

10.5.2 Code Definition

We define the code using the graph structure. The n bits of the codeword are associated with the vertices in L. The vertices in R act as parity checks.

Definition 10.14 (Tanner Code / Expander Code)**.** A vector $x \in \{0,1\}^n$ (on L) is a valid codeword if for every vertex $v \in R$, the parity of its neighbors is zero:

$$\bigoplus_{u \in N(v)} x_u = 0.$$

This is a **Linear Code**. For linear codes, the minimum distance is equal to the minimum weight of any non-zero codeword.

Theorem 10.15. *The minimum distance of this code is at least αn.*

Proof. Suppose not. Let x be a non-zero codeword with weight $\|x\|_1 < \alpha n$. Let $S \subseteq L$ be the support of x (the positions where $x_i = 1$). $|S| < \alpha n$.

By the expansion property, $|N(S)| \geq \frac{3d}{4}|S|$.

The total number of edges incident to S is $|S|d$. Consider the average number of edges from S that a vertex in $N(S)$ sees:

$$\text{Avg} = \frac{\text{Total edges from } S}{|N(S)|} \leq \frac{|S|d}{\frac{3d}{4}|S|} = \frac{4}{3}.$$

Since the average degree (into S) is $4/3$, there must exist a vertex $v \in N(S)$ that sees exactly one edge from S. (If all vertices saw ≥ 2 edges, the average would be ≥ 2).

But this vertex v cannot satisfy its parity check. Exactly one of its neighbors (in S) is 1, and the rest (not in S) are 0. The parity is 1. This contradicts the assumption that x is a codeword.

Therefore, the minimum weight (and thus minimum distance) is at least $\alpha n = \Omega(n)$. □

10.5.3 Decoding

We have a code with large distance. Can we decode efficiently? Yes, using a simple iterative algorithm.

Call a node $v \in R$ **unhappy** if its parity check is not satisfied.

Algorithm 6: Belief Propagation / Bit Flipping Decoder

6.1 **while** *there are unhappy nodes in R* **do**
6.2 Find a vertex $u \in L$ such that a majority of its neighbors $N(u) \subseteq R$ are unhappy.
6.3 Flip the bit x_u.

Analysis Sketch. It can be shown that this algorithm converges quickly.

- If we flip x_u, the number of unhappy neighbors of u decreases. This tends to decrease the total number of unhappy checks $U(x)$.

- We need to argue that a valid move always exists if $U(x) \neq 0$.
- We also need to show that if the received word is close enough to a codeword y, this process converges to y. (For the proofs, see many lecture notes online.)

The decoding is very fast. Expander-based codes (like Tornado Codes, related to LDPC codes) are very useful in practice.