Chapter 18

Random Walks V

By Sariel Har-Peled, December 30, 2015[1]

> "Is there anything in the Geneva Convention about the rules of war in peacetime?" Stanko wanted to know, crawling back toward the truck. "Absolutely nothing," Caulec assured him. "The rules of war apply only in wartime. In peacetime, anything goes."
>
> – Romain Gary, Gasp.

18.1. Rapid mixing for expanders

We remind the reader of the following definition of expander.

Definition 18.1.1. Let $\mathsf{G} = (V, E)$ be an undirected d-regular graph. The graph G is a (n, d, c)-***expander*** (or just c-***expander***), for every set $S \subseteq V$ of size at most $|V|/2$, there are at least $cd|S|$ edges connecting S and $\overline{S} = V \setminus S$; that is $e\left(S, \overline{S}\right) \geq cd|S|$,

Guaranteeing aperiodicity Let G be a (n, d, c)-expander. We would like to perform a random walk on G. The graph G is connected, but it might be periodic (i.e., bipartite). To overcome this, consider the random walk on G that either stay in the current state with probability $1/2$ or traverse one of the edges. Clearly, the resulting Markov Chain (MC) is aperiodic. The resulting *transition matrix* is

$$\mathsf{Q} = \mathsf{M}/2d + \mathcal{I}/2,$$

where M is the adjacency matrix of G and $\mathcal{I}$ is the identity $n \times n$ matrix. Clearly Q is doubly stochastic. Furthermore, if $\widehat{\lambda_i}$ is an eigenvalue of M, with eigenvector v_i, then

$$\mathsf{Q}v_i = \frac{1}{2}\left(\frac{\mathsf{M}}{d} + \mathcal{I}\right)v_i = \frac{1}{2}\left(\frac{\widehat{\lambda_i}}{d} + 1\right)v_i.$$

As such, $\left(\widehat{\lambda_i}/d + 1\right)/2$ is an eigenvalue of Q. Namely, if there is a spectral gap in the graph G, there would also be a similar spectral gap in the resulting MC. This MC can be generated by adding to each vertex d self loops, ending up with a $2d$-regular graph. Clearly, this graph is still an expander if the original graph is an expander, and the random walk on it is aperiodic.

From this point on, we would just assume our expander is aperiodic.

18.1.1. Bounding the mixing time

For a MC with n states, we denote by $\pi = (\pi_1, \ldots, \pi_n)$ its stationary distribution. We consider only nicely behave MC that fall under Theorem 18.4.1$_{p6}$. As such, no state in the MC has zero stationary probability.

Definition 18.1.2. Let $\mathsf{q}^{(t)}$ denote the state probability vector of a Markov chain defined by a transition matrix Q at time $t \geq 0$, given an initial distribution $\mathsf{q}^{(0)}$. The *relative pairwise distance* of the Markov chain at time t is

$$\Delta(t) = \max_i \frac{\left|\mathsf{q}_i^{(t)} - \pi_i\right|}{\pi_i}.$$

Namely, if $\Delta(t)$ approaches zero then $\mathsf{q}^{(t)}$ approaches π.

We remind the reader that we saw a construction of a constant degree expander with constant expansion. In its transition matrix Q, we have that $\widehat{\lambda_1} = 1$, and $-1 \leq \widehat{\lambda_2} < 1$, and furthermore the *spectral gap* $\widehat{\lambda_1} - \widehat{\lambda_2}$ was a constant (the two properties are equivalent, but we proved only one direction of this).

We need a slightly stronger property (that does hold for our expander construction). We have that $\widehat{\lambda_2} \geq \max_{i=2}^{n} \left|\widehat{\lambda_i}\right|$.

Theorem 18.1.3. *Let Q be the transition matrix of an aperiodic (n, d, c)-expander. Then, for any initial distribution $\mathsf{q}^{(0)}$, we have that*

$$\Delta(t) \leq n^{3/2}\left(\widehat{\lambda_2}\right)^t.$$

Namely, since $\widehat{\lambda_2}$ is a constant smaller than 1, the distance $\Delta(t)$ drops exponentially with t.

Proof: We have that $\mathsf{q}^{(t)} = \mathsf{q}^{(0)}\mathsf{Q}^t$. Let $\mathcal{B}(\mathsf{Q}) = \langle v_1, \ldots, v_n \rangle$ denote the orthonormal eigenvector basis of Q (see Definition 18.4.2$_{p6}$), and write $\mathsf{q}^{(0)} = \sum_{i=1}^{n} \alpha_i v_i$. Since $\widehat{\lambda_1} = 1$, we have that

$$\mathsf{q}^{(t)} = \mathsf{q}^{(0)}\mathsf{Q}^t = \sum_{i=1}^{n} \alpha_i\left(v_i \mathsf{Q}^t\right) = \sum_{i=1}^{n} \alpha_i\left(\widehat{\lambda_i}\right)^t v_i = \alpha_1 v_1 + \sum_{i=2}^{n} \alpha_i\left(\widehat{\lambda_i}\right)^t v_i.$$

Since $v_1 = \left(1/\sqrt{n}, 1/\sqrt{n}, \ldots, 1/\sqrt{n}\right)$, and $\left|\widehat{\lambda_i}\right| \leq \widehat{\lambda_2} < 1$, for $i > 1$, we have that $\lim_{t\to\infty}\left(\widehat{\lambda_i}\right)^t = 0$, and thus

$$\pi = \lim_{t\to\infty} \mathsf{q}^{(t)} = \alpha_1 v_1 + \sum_{i=2}^{n} \alpha_i\left(\lim_{t\to\infty}\left(\widehat{\lambda_i}\right)^t\right) v_i = \alpha_1 v_1.$$

Now, since $v_1, \ldots, v_n$ is an orthonormal basis, and $\mathsf{q}^{(0)} = \sum_{i=1}^{n} \alpha_i v_i$, we have that $\left\|\mathsf{q}^{(0)}\right\|_2 = \sqrt{\sum_{i=1}^{n} \alpha_i^2}$. Thus implies that

$$\begin{aligned}
\left\|\mathsf{q}^{(t)} - \pi\right\|_1 &= \left\|\mathsf{q}^{(t)} - \alpha_1 v_1\right\|_1 = \left\|\sum_{i=2}^{n} \alpha_i\left(\widehat{\lambda_i}\right)^t v_i\right\|_1 \leq \sqrt{n}\left\|\sum_{i=2}^{n} \alpha_i\left(\widehat{\lambda_i}\right)^t v_i\right\|_2 = \sqrt{n}\sqrt{\sum_{i=2}^{n}\left(\alpha_i\left(\widehat{\lambda_i}\right)^t\right)^2} \\
&\leq \sqrt{n}\left(\widehat{\lambda_2}\right)^t \sqrt{\sum_{i=2}^{n}(\alpha_i)^2} \leq \sqrt{n}\left(\widehat{\lambda_2}\right)^t \left\|\mathsf{q}^{(0)}\right\|_2 \leq \sqrt{n}\left(\widehat{\lambda_2}\right)^t \left\|\mathsf{q}^{(0)}\right\|_1 = \sqrt{n}\left(\widehat{\lambda_2}\right)^t,
\end{aligned}$$

since $\mathsf{q}^{(0)}$ is a distribution. Now, since $\pi_i = 1/n$, we have

$$\Delta(t) = \max_i \frac{\left|\mathsf{q}_i^{(t)} - \pi_i\right|}{\pi_i} = \max_i n\left|\mathsf{q}_i^{(t)} - \pi_i\right| \leq n \max_i \left\|\mathsf{q}^{(t)} - \pi\right\|_1 \leq n\sqrt{n}\left(\widehat{\lambda_2}\right)^t. \qquad \blacksquare$$

18.2. Probability amplification by random walks on expanders

We are interested in performing probability amplification for an algorithm that is a **BPP** algorithm (see Definition 18.4.3). It would be convenient to work with an algorithm which is already somewhat amplified. That is, we assume that we are given a **BPP** algorithm **Alg** for a language L, such that

(A) If $x \in L$ then $\mathbf{Pr}\Big[\mathbf{Alg}(x) \text{ accepts}\Big] \geq 199/200$.

(B) If $x \notin L$ then $\mathbf{Pr}\Big[\mathbf{Alg}(x) \text{ accepts}\Big] \leq 1/200$.

We assume that **Alg** requires a random bit string of length n. So, we have a constant degree expander G (say of degree d) that has at least $200 \cdot 2^n$ vertices. In particular, let

$$U = |V(\mathsf{G})|,$$

and since our expander construction grow exponentially in size (but the base of the exponent is a constant), we have that $U = O(2^n)$. (Translation: We can not quite get an expander with a specific number of vertices. Rather, we can guarantee an expander that has more vertices than we need, but not many more.)

We label the vertices of G with all the binary strings of length n, in a round robin fashion (thus, each binary string of length n appears either $\left\lceil |V(\mathsf{G})|/2^n \right\rceil$ or $\left\lfloor |V(\mathsf{G})|/2^n \right\rfloor$ times). For a vertex $v \in V(\mathsf{G})$, let $\mathsf{s}(v)$ denote the binary string associated with v.

Consider a string x that we would like to decide if it is in L or not. We know that at least $99/100U$ vertices of G are labeled with "random" strings that would yield the right result if we feed them into **Alg** (the constant here deteriorated from $199/200$ to $99/100$ because the number of times a string appears is not identically the same for all strings).

The algorithm. We perform a random walk of length $\mu = \alpha\beta k$ on G, where α and β are constants to be determined shortly, and k is a parameter. To this end, we randomly choose a starting vertex X_0 (this would require $n + O(1)$ bits). Every step in the random walk, would require $O(1)$ random bits, as the expander is a constant degree expander, and as such overall, this would require $n + O(k)$ random bits.

Now, lets $X_0, X_1, \ldots, X_\mu$ be the resulting random walk. We compute the result of

$$Y_i = \mathbf{Alg}(x, r_i), \qquad \text{for} \qquad i = 0, \ldots, \nu, \qquad \text{and } \nu = \alpha k,$$

where $r_i = \mathsf{s}\Big(X_{i \cdot \beta}\Big)$. Specifically, we use the strings associated with nodes that are in distance β from each other along the path of the random walk. We return the majority of the bits $Y_0, \ldots, Y_{\alpha k}$ as the decision of whether $x \in L$ or not.

We assume here that we have a *fully explicit* construction of an expander. That is, given a vertex of an expander, we can compute all its neighbors in polynomial time (in the length of the index of the vertex). While the construction of expander shown is only explicit it can be made fully explicit with more effort.

18.2.1. The analysis

Intuition. Skipping every β nodes in the random walk corresponds to performing a random walk on the graph G^β; that is, we raise the graph to power k. This new graph is a much better expander (but the degree had deteriorated). Now, consider a specific input x, and mark the bad vertices for it in the graph G. Clearly, we mark at most $1/100$ fraction of the vertices. Conceptually, think about these vertices as being uniformly spread in the graph and far apart. From the execution of the algorithm to fail, the random walk needs to visit $\alpha k/2$ bad vertices in the random walk in G^k. However, the probability for that is extremely small - why would the random walk keep stumbling into bad vertices, when they are so infrequent?

The real thing. Let Q be the transition matrix of G. We assume, as usual, that the random walk on G is aperiodic (if not, we can easily fix it using standard tricks), and thus ergodic. Let $\mathsf{B} = \mathsf{Q}^\beta$ be the transition matrix of the random walk of the states we use in the algorithm. Note, that the eigenvalues (except the first one) of B "shrink". In particular, by picking β to be a sufficiently large constant, we have that

$$\widehat{\lambda_1}(\mathsf{B}) = 1 \quad \text{and} \quad \left|\widehat{\lambda_i}(\mathsf{B})\right| \leq \frac{1}{10}, \qquad \text{for } i = 2, \ldots, U.$$

For the input string x, let W be the matrix that has 1 in the diagonal entry W_{ii}, if and only $\mathbf{Alg}(x, \mathsf{s}(i))$ returns the right answer, for $i = 1, \ldots, U$. (We remind the reader that $\mathsf{s}(i)$ is the string associated with the ith vertex, and $U = |V(\mathsf{G})|$.) The matrix W is zero everywhere else. Similarly, let $\overline{\mathsf{W}} = \mathcal{I} - \mathsf{W}$ be the "complement" matrix having 1 at $\overline{\mathsf{W}}_{ii}$ iff $\mathbf{Alg}(x, \mathsf{s}(i))$ is incorrect. We know that W is a $U \times U$ matrix, that has at least $(99/100)U$ ones on its diagonal.

Lemma 18.2.1. *Let Q be a symmetric transition matrix, then all its eigenvalues of Q are in the range $[-1, 1]$.*

Proof: Let $p \in \mathbb{R}^n$ be an eigenvector with eigenvalue λ. Let p_i be the coordinate with the maximum absolute value in p. We have that

$$\left|\lambda p_i\right| = \left|(p\mathsf{Q})_i\right| = \left|\sum_{j=1}^{U} p_j \mathsf{Q}_{ji}\right| \leq \sum_{j=1}^{U} \left|p_j\right| \left|\mathsf{Q}_{ji}\right| \leq |p_i| \sum_{j=1}^{U} \left|\mathsf{Q}_{ji}\right| = \left|p_i\right|.$$

This implies that $|\lambda| \leq 1$.

(We used the symmetry of the matrix, in implying that Q eigenvalues are all real numbers.) ■

Lemma 18.2.2. *Let Q be a symmetric transition matrix, then for any $p \in \mathbb{R}^n$, we have that $\|p\mathsf{Q}\|_2 \leq \|p\|_2$.*

Proof: Let $\mathcal{B}(\mathsf{Q}) = \langle v_1, \ldots, v_n \rangle$ denote the orthonormal eigenvector basis of Q, with eigenvalues $1 = \lambda_1, \ldots, \lambda_n$. Write $p = \sum_i \alpha_i v_i$, and observe that

$$\left\|p\mathsf{Q}\right\|_2 = \left\|\sum_i \alpha_i v_i \mathsf{Q}\right\|_2 = \left\|\sum_i \alpha_i \lambda_i v_i\right\|_2 = \sqrt{\sum_i \alpha_i^2 \lambda_i^2} \leq \sqrt{\sum_i \alpha_i^2} = \left\|p\right\|_2,$$

since $|\lambda_i| \leq 1$, for $i = 1, \ldots, n$, by Lemma 18.2.1. ■

Lemma 18.2.3. *Let $\mathsf{B} = \mathsf{Q}^\beta$ be the transition matrix of the graph G^β. For all vectors $p \in \mathbb{R}^n$, we have: (i) $\|p\mathsf{B}\mathsf{W}\|_2 \leq \|p\|_2$, and (ii) $\left\|p\mathsf{B}\overline{\mathsf{W}}\right\| \leq \|p\| /5$.*

Proof: (i) Since multiplying a vector by W has the effect of zeroing out some coordinates, its clear that it can not enlarge the norm of a matrix. As such, $\|p\mathsf{B}\mathsf{W}\|_2 \leq \|p\mathsf{B}\|_2 \leq \|p\|_2$ by Lemma 18.2.2.

(ii) Write $p = \sum_i \alpha_i v_i$, where $v_1, \ldots, v_n$ is the orthonormal basis of Q (and thus also of B), with eigenvalues $1 = \widehat{\lambda_1}, \ldots, \widehat{\lambda_n}$. We remind the reader that $v_1 = (1, 1, \ldots, 1)/\sqrt{n}$. Since $\overline{\mathsf{W}}$ zeroes out at least 99/100 of the entries of a vectors it is multiplied by (and copy the rest as they are), we have that $\left\|v_1\overline{\mathsf{W}}\right\| \leq \sqrt{(n/100)(1/\sqrt{n})^2} \leq 1/10 = \|v_1\|/10$. Now, for any $x \in \mathbb{R}^U$, we have $\left\|x\overline{\mathsf{W}}\right\| \leq \|x\|$. As such, we have that

$$\left\|p\mathsf{B}\overline{\mathsf{W}}\right\|_2 = \left\|\sum_i \alpha_i v_i \mathsf{B}\overline{\mathsf{W}}\right\|_2 \leq \left\|\alpha_1 v_1 \mathsf{B}\overline{\mathsf{W}}\right\| + \left\|\sum_{i=2}^{U} \alpha_i v_i \mathsf{B}\overline{\mathsf{W}}\right\|$$

$$\leq \left\|\alpha_1 v_1 \overline{\mathsf{W}}\right\| + \left\|\left(\sum_{i=2}^{U} \alpha_i v_i \widehat{\lambda}_i^{\beta}\right)\overline{\mathsf{W}}\right\| \leq \frac{|\alpha_1|}{10} + \left\|\sum_{i=2}^{U} \alpha_i v_i \widehat{\lambda}_i^{\beta}\right\|$$
$$\leq \frac{|\alpha_1|}{10} + \sqrt{\sum_{i=2}^{U}\left(\alpha_i \widehat{\lambda}_i^{\beta}\right)^2} \leq \frac{|\alpha_1|}{10} + \frac{1}{10}\sqrt{\sum_{i=2}^{U}\alpha_i^2} \leq \frac{\|p\|}{10} + \frac{1}{10}\|p\| \leq \frac{\|p\|}{5},$$

since $\left|\lambda_i^{\beta}\right| \leq 1/10$, for $i = 2, \ldots, n$. ■

Consider the strings $r_0, \ldots, r_\nu$. For each one of these strings, we can write down whether its a "good" string (i.e., **Alg** return the correct result), or a bad string. This results in a binary pattern $b_0, \ldots, b_k$. Given a distribution $p \in \mathbb{R}^U$ on the states of the graph, its natural to ask what is the probability of being in a "good" state. Clearly, this is the quantity $\|p\mathsf{W}\|_1$. Thus, if we are interested in the probability of a specific pattern, then we should start with the initial distribution p^0, truncate away the coordinates that represent an invalid state, apply the transition matrix, again truncate away forbidden coordinates, and repeat in this fashion till we exhaust the pattern. Clearly, the ℓ_1-norm of the resulting vector is the probability of this pattern. To this end, given a pattern $b_0, \ldots, b_k$, let $\mathcal{S} = \langle S_0, \ldots, S_\nu \rangle$ denote the corresponding sequence of "truncating" matrices (i.e., S_i is either W or $\overline{\mathsf{W}}$). Formally, we set $S_i = \mathsf{W}$ if **Alg**(x, r_i) returns the correct answer, and set $S_i = \overline{\mathsf{W}}$ otherwise.

The above argument implies the following lemma.

Lemma 18.2.4. *For any fixed pattern $b_0, \ldots, b_\nu$ the probability of the random walk to generate this pattern of random strings is $\left\|p^{(0)} S_0 \mathsf{B} S_1 \ldots \mathsf{B} S_\nu\right\|_1$, where $\mathcal{S} = \langle S_0, \ldots, S_\nu \rangle$ is the sequence of W and $\overline{\mathsf{W}}$ encoded by this pattern.*

Theorem 18.2.5. *The probability that the majority of the outputs **Alg**(x, r_0), **Alg**(x, r_1), $\ldots$, **Alg**(x, r_k) is incorrect is at most $1/2^k$.*

Proof: The majority is wrong, only if (at least) half the elements of the sequence $\mathcal{S} = \langle S_0, \ldots, S_\nu \rangle$ belong to $\overline{\mathsf{W}}$. Fix such a "bad" sequence $\mathcal{S}$, and observe that the distributions we work with are vectors in $\mathbb{R}^U$. As such, if p^0 is the initial distribution, then we have that

$$\mathbf{Pr}\left[\mathcal{S}\right] = \left\|p^{(0)} S_0 \mathsf{B} S_1 \ldots \mathsf{B} S_\nu\right\|_1 \leq \sqrt{U}\left\|p^{(0)} S_0 \mathsf{B} S_1 \ldots \mathsf{B} S_\nu\right\|_2 \leq \sqrt{U}\frac{1}{5^{\nu/2}}\left\|p^{(0)}\right\|_2,$$

by Lemma 18.3.1 below (i.e., Cauchy-Schwarz inequality) and by repeatedly applying Lemma 18.2.3, since half of the sequence $\mathcal{S}$ are $\overline{\mathsf{W}}$, and the rest are W. The distribution $p^{(0)}$ was uniform, which implies that $\left\|p^{(0)}\right\|_2 = 1/\sqrt{U}$. As such, let $\mathcal{S}$ be the set of all bad patterns (there are $2^{\nu-1}$ such "bad" patterns). We have

$$\mathbf{Pr}\Big[\text{ majority is bad }\Big] \leq 2^k \sqrt{U}\frac{1}{5^{\nu/2}}\left\|p^{(0)}\right\|_2 = (4/5)^{\nu/2} = (4/5)^{\alpha k/2} \leq \frac{1}{2^k},$$

for $\alpha = 7$. ■

18.3. Some standard inequalities

Lemma 18.3.1. *For any vector $\mathbf{v} = (v_1, \ldots, v_d) \in \mathbb{R}^d$, we have that $\|\mathbf{v}\|_1 \leq \sqrt{d}\,\|\mathbf{v}\|_2$.*

Proof: We can safely assume all the coordinates of $\mathbf{v}$ are positive. Now,

$$\|\mathbf{v}\|_1 = \sum_{i=1}^{d} v_i = \sum_{i=1}^{d} v_i \cdot 1 = |\mathbf{v} \cdot (1, 1, \ldots, 1)| \leq \sqrt{\sum_{i=1}^{d} v_i^2}\sqrt{\sum_{i=1}^{d} 1^2} = \sqrt{d}\,\|\mathbf{v}\|,$$

by the Cauchy-Schwarz inequality. ■

18.4. Tools from previous lecture

Theorem 18.4.1 (Fundamental theorem of Markov chains). *Any irreducible, finite, and aperiodic Markov chain has the following properties.*

(i) All states are ergodic.

(ii) There is a unique stationary distribution π such that, for $1 \leq i \leq n$, we have $\pi_i > 0$.

(iii) For $1 \leq i \leq n$, we have $\mathtt{f}_{ii} = 1$ and $\mathtt{h}_{ii} = 1/\pi_i$.

(iv) Let $N(i,t)$ be the number of times the Markov chain visits state i in t steps. Then

$$\lim_{t \to \infty} \frac{N(i,t)}{t} = \pi_i.$$

Namely, independent of the starting distribution, the process converges to the stationary distribution.

Definition 18.4.2. Given a random walk matrix Q associated with a d-regular graph, let $\mathcal{B}(\mathsf{Q}) = \langle v_1, \ldots, v_n \rangle$ denote the *orthonormal eigenvector basis* defined by Q. That is, $v_1, \ldots, v_n$ is an orthonormal basis for $\mathbb{R}^n$, where all these vectors are eigenvectors of Q and $v_1 = 1^n/\sqrt{n}$. Furthermore, let $\widehat{\lambda_i}$ denote the ith eigenvalue of Q, associated with the eigenvector v_i, such that $\widehat{\lambda_1} \geq \widehat{\lambda_2} \geq \cdots \geq \widehat{\lambda_n}$.

Definition 18.4.3. The class **BPP** (for Bounded-error Probabilistic Polynomial time) is the class of languages that have a randomized algorithm **Alg** with worst case polynomial running time such that for any input $x \in \Sigma^*$, we have

(i) If $x \in L$ then $\mathbf{Pr}[\mathbf{Alg}(x) \text{ accepts}] \geq 3/4$.

(ii) If $x \notin L$ then $\mathbf{Pr}[\mathbf{Alg}(x) \text{ accepts}] \leq 1/4$.