1 Conductance

For a graph to have a small mixing time, we would like a random walk that starts within some small subset of nodes to quickly have a non-zero probability of being anywhere on the graph. To capture this idea, we define the notion of conductance as follows:

**Definition 1 (Conductance, first attempt)** Let $G = (V, E)$ be an undirected graph and $S \subset V$ be a set of nodes. Then the conductance $\Phi(S, \bar{S})$ is defined as

$$\Phi(S, \bar{S}) = \frac{|E_{S, \bar{S}}|}{|E_S|},$$

where

- $\bar{S} = V - S$,
- $E_S = \{(u, v) \in E \mid u, v \in S\}$ and
- $E_{S, \bar{S}} = \{(u, v) \in E \mid u \in S, v \in \bar{S}\}$.

The conductance of the graph $\Phi_G$ is defined as

$$\Phi_G = \min_{|S| \leq |V|/2} \Phi(S, \bar{S}).$$

To see why a graph with large conductance should have a small mixing time, let $S$ be the set of ‘overweight’ nodes $v$ such that $\pi_v > \tilde{\pi}_v$. Since $G$ has a large conductance, there are many ways for a random walker on $S$ to cross over to $\bar{S}$ and reduce the probability gap. In the extreme case where $\pi_v = 1/|S|$ for $v \in S$ and zero otherwise, then the probability of crossing the cut is precisely the conductance $\Phi(S, \bar{S})$.

The definition of $\Phi_G$ restricts the minimum to subsets $S$ of at most $|V|/2$ vertices to make sure that our results are not skewed by overly large sets. For example, consider $S = V - \{v\}$ when $G$ is $d$-regular: clearly, $\Phi(S, \{v\}) = d/[(n - 1)d] = 1/(n - 1)$, which is very small regardless of the large-scale properties of the graph. To get around this problem, we only compute conductances between subsets that form a constant fraction of the entire graph (the choice of the value $1/2$ is arbitrary).

In order to avoid this unnatural restriction, as well as to make the conductance symmetric with respect to cuts (so that $\Phi(S, \bar{S}) = \Phi(\bar{S}, S)$), we shall henceforth use a somewhat different definition:

**Definition 1’ (Conductance)** Let $G = (V, E)$ and $S$ be defined as in definition 1. Then the conductance of the cut $(S, \bar{S})$ is defined as

$$\Phi_S = \Phi_{\bar{S}} = \frac{|E_{S, \bar{S}}||E|}{|E_S||E_{\bar{S}}|},$$

and the graph conductance $\Phi_G$ is defined as the minimum conductance over all cuts.

Without loss of generality, suppose $|E_S| \leq |E_{\bar{S}}|$. But $E = E_S \cup E_{\bar{S}}$, so that $|E|/|E_S| \leq 2$. This implies that the new definition differs from the old one by a factor of at most 2.
**Definition 2 (L2-Distance)** The \( L_2 \)-distance between two distributions \( D_1 \) and \( D_2 \) over a discrete set \( X \) is denoted by \( \| D_1 - D_2 \|_2 \) and is defined as

\[
\| D_1 - D_2 \|_2 = \sqrt{\sum_{x \in X} (D_1(x) - D_2(x))^2}
\]

We are usually interested in the \( L_1 \)-distance between probability distributions, and the following lemma relates the two notions of distance:

**Lemma 3** Let \( D_1 \) and \( D_2 \) be two distributions. Then

\[
\| D_1 - D_2 \|_2 \leq \| D_1 - D_2 \|_1 \leq \sqrt{n} \| D_1 - D_2 \|_2
\]

**Proof** Write \( D = D_1 - D_2 \) and \( D(x) = D_1(x) - D_2(x) \). Then, on one hand,

\[
\| D \|_1^2 = \left( \sum_{x \in X} |D(x)| \right)^2 = \sum_{x \in X} D(x)^2 + \sum_{x \neq y} |D(x)||D(y)| \geq \sum_{x \in X} D(x)^2 = \| D \|_2^2.
\]

On the other hand, if we apply Chebychev’s sum inequality to the numbers \( |D(x_1)|, |D(x_2)|, \ldots, |D(x_n)| \) and 1, 1, \ldots, 1, then we get

\[
\left( \sum_{x \in X} |D(x)| \right)^2 \leq n \sum_{x \in X} |D(x)|^2,
\]

or \( \| D \|_1^2 \leq n \| D \|_2^2 \). Taking the square roots of these two inequalities, we have the result.

The following theorem (which we shall prove in a subsequent lecture) gives a precise relationship between the conductance of a graph and the mixing time:

**Theorem 4** Let \( P \) be the transition matrix corresponding to a random walk on a graph \( G \), and define \( d(t) = \| P^t \pi_0 - \tilde{\pi} \|_2^2 \) to be the square of the \( L_2 \)-distance between the distribution after \( t \) steps and the stationary distribution. Then

\[
d(t) \leq \left[ 1 - \frac{\Phi_G^2}{4} \right]^t d(0)
\]

Notice that \( d(0) \leq 2 \) for all starting distributions \( \pi_0 \), because

\[
\| \pi_0 - \tilde{\pi} \|_2 \leq \| \pi_0 \|_2 + \| \tilde{\pi} \|_2 \leq \| \pi_0 \|_1 + \| \tilde{\pi} \|_1 = 2.
\]

Therefore, if we set \( t = (4/\Phi_G^2) \ln(2n/\varepsilon^2) \), then

\[
d(t) \leq \left[ 1 - \frac{\Phi_G^2}{4} \right]^t \frac{4}{\Phi_G^2} \ln \frac{2n}{\varepsilon^2} \cdot 2 \leq \frac{\varepsilon^2}{n}
\]

by theorem 4. We can now apply lemma 3 to translate this into an \( L_1 \) bound:

\[
\| P^t \pi_0 - \tilde{\pi} \|_1 \leq \sqrt{n} \| P^t \pi_0 - \tilde{\pi} \|_2 = \sqrt{nd(t)} \leq \varepsilon.
\]

This formalizes our earlier intuition that a graph with a large conductance mixes fast. More specifically, it suffices to show that \( \Phi_G = \Omega(1/ \log n) \) to prove rapid mixing. In some cases, we can even show a constant lower bound on the conductance!
We shall be particularly interested in graphs that are $d$-regular for some $d$. In this case, the conductance is given by

$$
\Phi_G = \min_S \frac{|E_{S,\bar{S}}||E|}{|E_S||E_{\bar{S}}|} = \min_S \frac{|E_{S,\bar{S}}|d|V|}{d|S||\bar{S}|} = \frac{1}{d} \left( \min_S \frac{|E_{S,\bar{S}}||V|}{|S||\bar{S}|} \right).
$$

The parenthetized term above has a special name: it is the edge magnification $\mu$:

$$
\mu = \min_S \frac{|E_{S,\bar{S}}||V|}{|S||\bar{S}|},
$$

and for $d$-regular graphs, $\Phi_G = \mu/d$.

One important technique for lower-bounding the conductance of a graph is the method of canonical paths, which we have already used for the hypercube. The idea is to carefully choose a set of paths between every pair of nodes, such that no edge in the graph has too many paths going through it.

**Definition 5 (Congestion)** Let $P = \{p_{uv}\}$ be a set of canonical paths for a graph $G = (V, E)$, where $p_{uv}$ connects vertex $u$ to vertex $v$. Then the congestion of an edge $e \in E$ is defined as the number of paths $p \in P$ that use $e$. Also, the congestion of $G$ is defined as the maximum congestion over all edges $e$.

The congestion of a graph can be as large as $O(n^2)$—consider, for example, the line on $n$ nodes—but for many graphs, it is possible to find a set of canonical paths that makes the congestion small. For a graph of low conductance, however, there are bottleneck edges which must be congested by any chosen set of paths.

**Claim 6** If $G$ has congestion $\alpha n$ with respect to some set of canonical paths, then $\mu \geq 1/\alpha$.

**Proof** Fix a cut $(S, \bar{S})$ of $G$. Then the number of canonical paths $p_{uv}$ connecting $u \in S$ to $v \in \bar{S}$ is $|S||\bar{S}|$. Each of these paths has to use at least one edge $e$ in the cut, i.e., $e \in E_{S,\bar{S}}$. By the definition of congestion, we have

$$
\text{# of paths crossing cut} \leq \sum_{e \in E_{S,\bar{S}}} (\text{# of paths crossing } e) \leq |E_{S,\bar{S}}| \max_{E_{S,\bar{S}}} (\text{# of paths crossing } e) \leq |S||\bar{S}| \leq |E_{S,\bar{S}}|\alpha n \geq \frac{1}{\alpha}
$$

for all cuts $(S, \bar{S})$. The edge expansion $\mu$ is the minimum value of the left hand side of the above inequality, so $\mu \geq 1/\alpha$. 

Recall that in lecture 7 (weakly learning monotone functions), we studied the conductance of the hypercube on $n = 2^N$ nodes using canonical paths. We chose paths which had the property that an edge on a path, along with $N$ additional bits (or a complementary point), completely determined the start and end node (and therefore the path). This property, allowed us to argue that no more than $n$ distinct paths could pass through a given edge, bounding the congestion and hence the conductance. We will do something similar for the problem of uniformly generating graph matchings, which we shall address next.
2 Uniformly Generating Matchings

Given a bipartite graph \( G = (V, E) \) where \( m = |E| \), we wish to generate a matching of the vertices of the graph uniformly at random.\(^1\) We do this by constructing a Markov chain with states corresponding to matchings and in which transitions correspond to small local changes in the matching. Given an initial matching (state) \( M \), the possible transitions are defined as follows:

\[
\begin{align*}
\text{Pick an edge } e \in_R E \\
\text{if } e \in M, \\
&\quad \text{then set } M \leftarrow M \setminus \{e\} \\
\text{else if } M \cup \{e\} \text{ is a matching} \\
&\quad \text{then set } M \leftarrow M \cup \{e\} \\
\text{else} \\
&\quad \text{stay put}
\end{align*}
\]

The resulting Markov chain \( M = (S, T) \) has the following properties:

- It is \textit{undirected}, because every transition is reversible.
- It is \textit{connected}: to get from matching \( M_1 \) to \( M_2 \): Drop all the edges in \( M_1 \) to get to the empty matching, and then build up \( M_2 \) one edge at a time. In fact, this shows that the diameter of the chain is at most \( 2|M| \leq 2|V|/2 = |V| \), where \( M \) is a maximal matching.
- It is \textit{non-bipartite}, because it has at least one self-loop (for example, consider starting from any maximal matching and picking an edge not in the matching).
- It is \textit{regular} with degree \( m \), because for any initial matching, we can consider any of the \( m \) edges of \( G \) to add or remove.

In order to define the canonical paths on this Markov chain, we note that the symmetric difference \( M_1 \oplus M_2 \) of two matchings consists of a set of alternating paths and cycles. We fix an arbitrary ordering on the edges of \( G \), a start edge for every possible path or cycle, and a traversal direction for every cycle.

To convert \( M_1 \) into \( M_2 \), we consider the edges in \( M_1 \oplus M_2 \) in the order defined above. When we encounter an edge \( e \), we process the entire alternating path or cycle that contains it (as shown below). We keep doing this until there are no more paths or cycles to process.

- To process a path \( e_1 e_2 \ldots e_k \), we have to delete an edge before we can add a new one. Assume \( e_1 \) and \( e_k \) both must be added. If not, we can just delete them before running the algorithm. So \( k \) is odd. The algorithm is:

\[
\begin{align*}
i &\leftarrow 1 \\
\text{while } i \neq k \text{ do} \\
&\quad \text{Delete } e_{i+1} \\
&\quad \text{Insert } e_i \\
&\quad i \leftarrow i + 2 \\
&\quad \text{Insert } e_k
\end{align*}
\]

- To process a cycle \( e_1 e_2 \ldots e_k e_1 \), we need to be careful, because we must delete \textit{two} edges in the cycle before any insertions are possible. Assume \( e_1 \) must be deleted. Note that \( k \) must be even. The algorithm runs as follows:

\footnote{\( \)\text{It is possible to generate maximal and/or perfect matchings, but here we address the simpler problem of generating arbitrary matchings.}}

- To process a path \( e_1 e_2 \ldots e_k \), we have to delete an edge before we can add a new one. Assume \( e_1 \) and \( e_k \) both must be added. If not, we can just delete them before running the algorithm. So \( k \) is odd. The algorithm is:

\[
\begin{align*}
i &\leftarrow 1 \\
\text{while } i \neq k \text{ do} \\
&\quad \text{Delete } e_{i+1} \\
&\quad \text{Insert } e_i \\
&\quad i \leftarrow i + 2 \\
&\quad \text{Insert } e_k
\end{align*}
\]

- To process a cycle \( e_1 e_2 \ldots e_k e_1 \), we need to be careful, because we must delete \textit{two} edges in the cycle before any insertions are possible. Assume \( e_1 \) must be deleted. Note that \( k \) must be even. The algorithm runs as follows:
Given a transition \( e \in T \), we need to find a way to bound its congestion. We shall do so by answering the question: “what additional information do we need to reconstruct the endpoints of the path?” For the hypercube, we found this bound in terms of a number of bits, but in this case, we don’t even know how large \( S \) is. Luckily, however, claim 6, which bounds the conductance, requires the value of the congestion to be specified as a multiple of the chain size. Therefore, we shall specify the additional information in the form of another matching (the complementary point) and a small number of additional bits.

**Claim 7** Fix a transition \( M_a \rightarrow M_b \). We can reconstruct the starting and ending states \( M_1 \) and \( M_2 \) of the canonical path if we specify the additional information \( \bar{M} = (M_1 \oplus M_2) - M_a \).

**Proof** Using the ordering on edges, we can decide which edges in \( M_a \) have not yet been corrected. These edges must match \( M_1 \). The remaining edges of \( M_1 \) are given by the corrections contained in \( M_a \oplus \bar{M} \). Similarly, we can reconstruct \( M_2 \) as well. ■

Unfortunately, we are not quite done, because \( \bar{M} \) might not be a matching, so that it is unsuitable as a complementary point. However, it can be shown that we can always remove at most two edges from \( \bar{M} \) to make it into a matching. Therefore, it suffices to specify the resulting matching, along with one of \( m^2 \) possibilities for the two edges. This means that the edge congestion is at most \( m^2 |S| \).

By claim 6, \( \mu \geq 1/m^2 \). We have already noted that \( \mathcal{M} \) is \( m \)-regular, so that

\[
\Phi_G = \frac{\mu}{m} = \frac{1}{m^2} = \frac{1}{m^3}.
\]

The number of matchings is bounded by the number of subsets of the edge set, \( 2^m \). Using this, we can set

\[
t = 4 \frac{\Phi_G \ln 2|S|}{\varepsilon^2} \leq 4m^6 \ln \frac{2^{m+1}}{\varepsilon^2} = O(m^7 \ln(1/\varepsilon))
\]

to get within \( \varepsilon \) of the uniform distribution. Therefore, the Markov chain mixes rapidly, or in polynomial time.