# Lecture 15

*Lecture date: Oct 17* *Scribe: Amin Aminzadeh Gohari*

In this lecture, we complete the proof of the *bit-flipping algorithm* from lecture 14. Furthermore, we prove the *Gallager Bound*, an upper bound on the tolerable probability of noise under the assumption of a reliable communication. We start with a review of the definition of an expander graph and the description of the bit flipping algorithm.

## 1 Bit Flipping Algorithm

Assume that we have a binary linear code defined using a specific Code graph(or Tanner graph). A Code graph is a bipartite graph with the variable nodes on one side and check nodes on the other side. As we saw in the previous lectures, each check node imposes a parity constraint on the set of variable nodes connected to it. In the previous lecture, we defined a decoding algorithm which we claimed has a guarantee of success for all Code graphs which are expander and have fixed variable node degrees. Before making these definitions more precise, we review the definition of an expander graph:

**Definition 1** *A factor graph is a ($\epsilon$, $\delta$)-expander if for every subset $U$ of the set of variable nodes of size bounded by $|U| \leq \epsilon N$, it holds that $|\partial U| > \delta |U|$, where $\partial U$ is the set of factor nodes adjacent to at least one node in $U$.*

Assume that we have transmitted some codeword $W = (W_1, ..., W_m)$ and the decoder has received the output sequence $Y = (Y_1, ..., Y_m)$ over a BSC channel with parameter $p$ (meaning $P(Y_i = X_i) = 1 - p$).

The bit flipping algorithm starts out with sequence $Y = (Y_1, ..., Y_m)$. Each check node introduces a constraint. As we saw in the previous lecture, in each step, the algorithm tries to find a bit, flipping of which reduces the number of unsatisfied clauses. The algorithm stops when for every bit, the number of unsatisfied constraints is less or equal to the number of satisfied constraints. Since the number of unsatisfied constraints reduces in each step, the algorithm has to stop. We will prove the following theorem:

***Theorem 1.****Consider a Code graph where all variable nodes have degree $l$, and the graph is ($\epsilon$, $\delta$)-expander. Then the bit- flipping algorithm will correct any pattern of at most $\frac{N\epsilon}{2}$ errors.*

***Proof:*** Without loss of generality we can assume the all zero codeword has been sent (since the code is linear). Thus the hamming weight of the received codeword equals to the

number of bits we have received in error. Let

- *$X(t)$ denote the string at the $t^{th}$ iteration of the algorithm. Since we start out with the received message, we have $X(0) = Y$.*
- *$W(t)$ denote the hamming weight of $X(t)$.*
- *$u(t)$ denote the number of unsatisfied check nodes.*
- *$s(t)$ denote the number of satisfied checks that are adjacent to at least one $X_i$ such that $X_i = 1$.*

Since the number of errors is at most $\frac{N\epsilon}{2}$, we have $w(0) \leq \frac{N\epsilon}{2}$.

It is clear from the bit-flipping algorithm that $u(t)$ is a decreasing function of $t$. The following two propositions thus imply the result stated at Theorem 1.

***Proposition 1.*** *Let $t^*$ be the iteration in which the algorithm stops, then $w(t^*)$ is either zero, or is greater than or equal to $N\epsilon$.*

***Proof:***Assume that $0 < w(t^*) < N\epsilon$. Consider the nodes which are one. Because of expansion, the number of check nodes adjacent to those nodes that are one is at least $\frac{3}{4}lw(t^*)$. Let set $T$ denote these set of check nodes.

The check nodes adjacent to variable nodes that are ones are either counted in $u(t)$ or $s(t)$. Therefore $|T| = u(t^*) + s(t^*) > \frac{3}{4}lw(t^*)$.

On the other hand, since every satisfied check is adjacent to at least two variable nodes with $X_i = 1$, the number of edges going out from the set $T$ is at least $u(t^*) + 2s(t^*)$. But since the degree of each variable node is $l$, the number of edges going out from set $T$ is exactly equal to $lw(t^*)$, and therefore we get $u(t^*) + 2s(t^*) \leq lw(t^*)$. This inequality together with $u(t^*) + s(t^*) > \frac{3}{4}lw(t^*)$, one can easily show, imply that $u(t^*) > \frac{1}{2}lw(t^*)$.

But $u(t^*) > \frac{1}{2}lw(t^*)$ implies that there exist at least one variable node with $X_i = 1$ that is adjacent to more than $\frac{l}{2}$ check nodes. This is a contradiction since this variable node is adjacent to more unsatisfied nodes than satisfied nodes and thus the algorithm couldn't have stopped at iteration $t^*$.

***Proposition 2.*** *The algorithm can not terminate with a string of weight greater than $N\epsilon$.*
***Proof:***It is enough to prove that there is no $t$ such that $w(t) = N\epsilon$. Assume otherwise that $w(t) = N\epsilon$. The above inequalities show that $u(t) > \frac{l}{2}w(t) = \frac{l}{2}N\epsilon$. On the other hand, $u(t) \leq u(0) = \frac{lN\epsilon}{2}$ which is a contradiction.

## 2 The Gallager Bound

In this section, we prove the *Gallager Bound*. We first start with the problem setup: Consider a linear code of rate $R$, based on a tanner graph with $(\Lambda, P)$ degree distribution. Assume that such code when passed through a BSC channel with cross-over probability $P_M$ could be reliably decoded at the decoder (that is probability of error goes to zero with block length converging to infinity). The Gallager Bound states:

***Theorem 2.*** *Let $P_G$ be the solution of the equation: $H(p) = (1-R)\sum_i P_i H(\frac{1-(1-2p)^i}{2})$. Then $P_M \leq P_G$.*

***Comment:*** *As we saw in previous lectures, for the LDPC codes and with high probability, we have $R = 1 - \frac{\Lambda'(1)}{P'(1)}$.*

***Proof:***Let $X = (X_1, ..., X_n)$ to be the transmitted codeword, $Y = (Y_1, ..., Y_n)$ to be the received string. Assume $S = (S_1, ..., S_m)$ to be the values of check nodes corresponding to $Y = (Y_1, ..., Y_n)$. (Thus, we are assuming that there are $m$ check nodes in the code graph). Assume that $X$ is chosen uniformly random from the set of possible codewords. We will use the following Lemmas which we will prove at the end of this section:

***Lemma 1.*** *We have $H(Y) = H(X) + H(S)$.*

***Lemma 2.*** *If $H(X|Y) \geq \delta N$, the probability of decoding error is lower bounded by a constant depending on $\delta$.*

***Lemma 3.*** *Let $C_1$, $C_2$,...,$C_i$ to be i, binary i.i.d. random variables distributed according to Bernoulli distribution $B(p)$, then their binary sum $C = C_1 + C_2 + ... + C_i \ (mod 2)$ is distributed according to $B(\frac{1-(1-2p)^i}{2})$.*

Now we have $H(X|Y) = H(X) + H(Y|X) - H(Y) = NR + NH(p_M) - H(Y) =^1 NR + NH(p_M) - NR - H(S) = NH(p_M) - H(S)$ where (1) is true because of Lemma 1. We can upper bound $H(S)$ by $\sum_{i=1}^m H(S_i)$. Since $S_i$ is a binary addition of some variable nodes, each of which are binary random variables with uncertainty $h(p_M)$, we can apply Lemma 2 to get $H(S_i) = h(\frac{1-(1-2p_M)^{m_i}}{2})$ where $m_i$ is the number of variable nodes connecting to $S_i$. The number of check nodes having $m_i = r$ is equal to $\frac{\Lambda'(1)}{P'(1)} N P_i$. Therefore $\sum_{i=1}^m H(S_i) = \Lambda'(1)P'(1)N\sum_i P_i H(\frac{1-(1-2p_M)^i}{2}) = N(1-R)\sum_i P_i H(\frac{1-(1-2p_M)^i}{2})$. Hence $H(X|Y) \geq NH(p_M) - N(1-R)\sum_i P_i H(\frac{1-(1-2p_M)^i}{2}) = N[H(p_M) - (1-R)\sum_i P_i H(\frac{1-(1-2p_M)^i}{2})]$

Now if $p_M < p_G$, the expression inside the brackets is positive, hence $H(X|Y)$ is lower bounded by some $\delta N$. But this is in contradiction with the result of Lemma 3.

Now we prove the above Lemmas:

***Proof of Lemma 1:*** Since random variable $S$ is a function of $Y$, we have $H(Y) = H(Y,S) = H(S) + H(Y|S)$. So, we need to prove that $H(Y|S) = H(X)$. But $H(Y|S) = E_s H(Y|S=s)$. It is enough to prove that $H(Y|S=s)$ is equal to $H(X)$ for every $s$. But the set of all possible $y$'s for which $S(Y=y) = s$ is nothing but a affine linear subspace of the whole space; i.e. $\{y : S(Y=y) = s\} = \{y : S(Y=y) = 0\} + y_0 = \{$ The Set of Codewords $\} + y_0$ where $y_0$ is some arbitrary string satisfying the property $S(Y=y_0) = s$. Moreover, because the symmetry of the linear codes, $Y|S = s$ would have a uniform distribution over the mentioned set. Hence $H(Y|S=s) = \log|\{y : S(Y=y) = s\}| = NR = H(X)$. Therefore $H(Y) = H(X) + H(S)$

Remark : *We have only use $H(Y) \leq H(X) + H(S)$ in the proof, which could be proved even easier: For every $s$, $H(Y|S=s) \leq \log|\{y : S(Y=y) = s\}| = NR = H(X)$.*

***Proof of Lemma 2:*** Since $H(X|Y) = E_y(H(X|Y=y))$, one can conclude that with probability at least $\frac{\delta}{2}$ over the observation of random variable $Y$, $H(X|Y=y) \geq \frac{\delta}{2}N$. Having received $Y = y$, the best choice for $X$ is the one that maximizes $P(X=x|Y=y)$. Assume that the maximum of $P(X=x|Y=y)$ is $v$. It is enough to prove that $v$ is not close to one (since then probability of error would be at least $(1-v)\frac{\delta}{2}$). In order to prove that $v$ is not close to one, we upper bound the entropy of $H(X|Y=y)$ by the entropy of the distribution on codewords which takes $x$ with probability $v$ and is uniform over all other codewords with total probability $(1-v)$. Therefore $H(X|Y=y)$ is upper bounded by $-v\log v - (2^{RN}-1).[\frac{1}{2^{RN}-1}(1-v)\log((2^{RN}-1)(1-v)] \leq -v\log(v) - (1-v)\log(1-v) + RN(1-v) \leq 1 + RN(1-v)$. Comparing the lower bound on $H(X|Y=y)$ and this upper bound, it can be seen that $v$ is bounded away from 1.

***Proof of Lemma 3:*** This can be easily proved using induction, or by defining the random variables $Q_i = (-1)^{C_i}$ and expanding $E(Q_1.Q_2...Q_i)$(which we can do since $Q_i$'s are independent). The binary addition of $C_i$'s is zero if an only if the product of $Q_i$'s is one.