

# lil' Lemma

C. Garfinkle

**Lemma 1.** Suppose  $E \subseteq 2^{[m]}$  is a hypergraph with the SIP and let  $d_1 \geq d_2 \geq \dots \geq d_m$  be the degree sequence of nodes in  $E$ . If for some  $\bar{m}$  the map  $\pi : E \rightarrow 2^{[\bar{m}]}$  has  $\sum_{S \in E} |\pi(S)| = \sum_{S \in E} |S|$  and:

$$|\cap \pi(F)| \leq |\cap F| \text{ for all } F \subseteq E, \quad (1)$$

then  $\bar{m} \geq \sum_i d_i/d_1$  and the association  $i \mapsto \cap \pi(F(i))$  defines an injective map from  $J$  to  $[\bar{m}]$  for some  $J \in \binom{[m]}{p}$ , where  $p$  (provided it exists) is the largest positive integer in  $\{1, \dots, m\}$  satisfying:

$$\sum_{i=\ell}^m d_i > (\bar{m} + 1 - \ell)(d_\ell - 1) \text{ for all } \ell \leq p. \quad (2)$$

If  $\bar{m} < \sum_i d_i/(d_1 - 1)$  then  $p \geq 1$ . In particular, if  $\bar{m} = m$  and  $E$  is regular, then  $\pi$  is induced by a permutation on  $[m]$ .

*Proof.* Consider the collection of pairs:  $T_1 := \{(i, S) : i \in \pi(S), S \in E\}$ , which number  $|T_1| = \sum_{S \in E} |\pi(S)| = \sum_{S \in E} |S| = \sum_{i \in [m]} d_i$ . Note that assumption (1) implies  $\bar{m} \geq |T_1|/d_1$ , since otherwise pigeonholing the elements of  $T_1$  with respect to their set of possible first indices  $[\bar{m}]$  would lead us to conclude that there are more than  $d_1$  sets in  $E$  sharing a common element.

By (2) we have  $|T_1| > \bar{m}(d_1 - 1)$ , which implies, again by the pigeonhole principle, that there must be at least  $d_1$  elements of  $T_1$  sharing the same first index. By (1), the intersection of the set  $F_1$  consisting of their second indices is non-empty. As  $d_1 \geq d_i$  for all  $i$ , and  $E$  satisfies the SIP, it must be that the sets in  $F_1$  intersect at a singleton. Since  $\cap \pi(F_1)$  is non-empty, applying (1) again implies  $\cap \pi(F_1) = \{i_1\}$  for some  $i_1 \in [m]$ .

If  $p = 1$  then we are done. Otherwise, define  $T_2 := T_1 \setminus \{(i_1, S) \in T_1 : S \in E\}$ , which contains  $|T_2| = |T_1| - d_1 = \sum_{i=2}^m d_i$  ordered pairs having  $\bar{m} - 1$  distinct first indices. By (2) we have  $|T_2| > (\bar{m} - 1)(d_2 - 1)$  and reiterating the above arguments produces a (necessarily) distinct index  $i_2$ . Iterating the arguments  $p$  times yields the set of singletons  $J = \{\cap F_1, \dots, \cap F_p\} \subseteq [m]$ .  $\square$

**Remark 1** (Regular hypergraphs). Note that if  $E$  is  $d$ -regular then  $\sum_{i=\ell}^m d = (m - \ell + 1)d$  and we have  $\bar{m} \geq \sum_i d_i/d_1 \geq m$  while (2) becomes  $\ell < \bar{m} - (\bar{m} - m)d + 1$  for all  $\ell \leq p$ . Hence, it suffices to know that  $p = \bar{m} - (\bar{m} - m)d$  is positive to know that (2) is satisfied, which is true if and only if  $\bar{m} < md/(d + 1)$ . Hence we must have  $m \leq \bar{m} < md/(d - 1)$  for the conclusion of the lemma to hold for some non-empty  $J \subseteq [m]$ . We therefore have  $J = [m]$  when  $\bar{m} = m$  and  $E$  is regular.

**Remark 2** (General hypergraphs). For what range of values of  $\bar{m}$  can we guarantee there exists some  $p \in \{1, \dots, m\}$  in general? If  $\bar{m} < \sum_i d_i/(d_1 - 1)$  then we can peel off at least one index for any hypergraph, i.e.  $p \geq 1$ . Can we find a lower bound on  $\bar{m}$  in general, though (not just for regular hypergraphs)? The difficulty is we have to make sure (2) holds for all values of  $\ell \leq p$ , which was easy for regular hypergraphs (by the resulting transitivity of (2)). My thought is that in general the lower bound  $\bar{m} \geq \sum_i d_i/d_1$  which \*has\* to hold given (1) might not correspond to what condition (2), which comes from our crappy pigeonholing, gives us as a constraint on  $\bar{m}$  when we make the substitution  $p = m$  (as was the case for regular hypergraphs). Could we have  $\bar{m} < m$  but still recover a positive  $p$  number of indices?

**Question 1.** Do we only need  $E$  to satisfy the SIP on the  $p$  nodes of highest degree, those we end up isolating, for all of Chaz' Thm. to run through? That would be good!