Iterated Rounding for Hypergraph Matching[1]

- In this lecture, we look at two simple applications of relaxation and rounding for *packing* problems. In these problems, we wish to *maximize* a linear function. We will use the fact that LP optima are basic feasible solutions, and how this helps in approximation algorithm design.

- ***Hypergraph Matching.*** A hypergraph $H = (V, E)$ is a generalization of graphs where every (hyper)edge $e \in V$ is an arbitrary subset of the vertices instead of a pair. A k-uniform hypergraph is one where $|e| = k$ for all $e \in E$. In the hypergraph matching problem, we need to find a collection $M \subseteq E$ such that for any two $e, e' \in M$ we have $e \cap e' = \emptyset$, and $|M|$ is as large as possible. We will focus on solving the problem on 3-uniform hypergraphs.

- *A Simple $\frac{1}{3}$-approximation.* We begin with a simple algorithm whose structure we will borrow for our final algorithm. Before stating it, let us make a useful definition: given an edge $e \in E$, let $N(e)$ denote all edges which intersect e. That is, $N(e) := \{f : f \cap e \neq \emptyset\}$. Here is the algorithm.

```
1: procedure MAXIMAL HYPERGRAPH MATCHING(3-uniform H = (V, E)):
2:     M ← ∅; F ← E.
3:     while F ≠ ∅ do:
4:         Pick an arbitrary edge e ∈ F and add it to M.
5:         Remove all edges in N(e) from F.
6:     return M.
```

Theorem 1. MAXIMAL HYPERGRAPH MATCHING is a $\frac{1}{3}$-approximation.

Proof. Suppose M^* is the largest cardinality matching. Let us consider two variables o and a. Initially $o \leftarrow \texttt{opt} = |M^*|$ and $a \leftarrow 0$. At every run of the while loop we increment $a \leftarrow a + 1$ to indicate how many edges are added to M. We also decrement o by the number of edges of M^* removed in Line 5. The crucial observation is that this number is *at most* 3. Why? Let $e = (u, v, w)$ be the edge added to M. In Line 5, we remove all edges in F incident to u, v, and w. Since M^* is a matching, there can be at most one edge $f_1 \in M^*$ containing u, at most one edge $f_2 \in M^*$ containing v, and at most one edge $f_3 \in M^*$ containing w. Thus, opt decrements by at most 3.

At the termination of the while loop, we have $a = \texttt{alg}$ indicating the number of while loops. We also have that the value of o at the end is at least $\texttt{opt} - 3\texttt{alg}$. But the value of o at the end *must* be 0 since there are no more edges left at the end. So, $\texttt{opt} - 3\texttt{alg} \leq 0$ implying $\texttt{alg} \geq \texttt{opt}/3$. □

[1] *Lecture notes by Deeparnab Chakrabarty. Last modified : 2nd Jan, 2022*
These have not gone through scrutiny and may contain errors. If you find any, or have any other comments, please email me at deeparnab@dartmouth.edu. Highly appreciated!

Exercise: *Suppose the hyperedges have weights $w(e)$ and the goal was to pick the largest* weight *matching. Modify the above algorithm and analysis to describe a $\frac{1}{3}$-algorithm for the same.*

- *LP Relaxation.* One can write the following natural LP relaxation for the 3-hypergraph matching problem.

$$\mathsf{lp}(H) := \text{maximize} \quad \sum_{e \in E} x_e \tag{1}$$
$$\sum_{e: v \in e} x_e \leq 1, \qquad \forall v \in V$$
$$x_e \geq 0, \qquad \forall e \in E$$

We can solve this LP and obtain a solution x_e for all $e \in E$. How can we use it to get a large matching? Which edge e should we pick in the matching? Note that once we pick e, we must delete all edges in $N(e)$ in the subsequent rounds. The total "fractional mass" lost, therefore, is $x(N(e)) := \sum_{f \in N(e)} x_f$. Thus, one idea is to pick an edge e with the "smallest" $x(N(e))$.

```
1: procedure LP-BASED HYPERGRAPH MATCHING(3-uniform H = (V, E)):
2:     M ← ∅; F ← E.
3:     while F ≠ ∅ do:
4:         Solve (1) on the residual hypergraph (V, F) to get an optimum bfs x.
5:         If there is an edge e with x_e = 0, remove it from F and break.
6:         Pick an edge e ∈ F with smallest x(N(e)) and add it to M.
7:         Remove all edges from N(e) from F.
8:     return M.
```

Theorem 2. The matching returned by the above algorithm is a $\frac{3}{7}$-approximation.

Proof. The proof follows the same structure as that of Theorem 1. We initialize the two variable o and a. If we run Line 5, then we keep the variables unchanged. Otherwise, we increment a by 1 in each iteration, and thus at the end $a = \mathsf{alg}$. We initialize o with $\mathsf{lp} := \mathsf{lp}(H)$. In every iteration, if D is the set of edges removed in Line 7, then we decrement o by $x(N(e)))$ where x is the solution to the LP obtained in Line 4. The heart of the analysis lies in the following claim.

Claim 1. At the beginning of Step Line 6, there exists an edge $e \in F$ with $x(N(e)) \leq \frac{7}{3}$.

We prove the above claim in the next bullet point. Right now, note that the above suffices to prove Theorem 2. Indeed, let lp_i be the value of the LP just before the ith iteration and let $\mathbf{x}^{(i)}$ be an optimal solution. If e is the edge picked in this iteration, then note that $\mathbf{x}'$ which just zeroes out $\mathbf{x}^{(i)}$ at $N(e)$ is a valid solution before the $i+1$th iteration. Thus, $\mathsf{lp}_{i+1} \geq \mathsf{lp}_i - x(N(e)) \geq \mathsf{lp}_i - \frac{7}{3}$. On the other hand, the LP-value at the end of the algorithm must be 0 since all edges are deleted. Therefore, if the algorithm runs for alg rounds, we have $0 \geq \mathsf{lp} - \frac{7\mathsf{alg}}{3}$, proving the theorem. □

- *Proof of Claim 1*. The proof of the claim will use the fact that x was a basic feasible solution. Recall that a basic feasible solution satisfies dimension-many linearly independent inequalities as equality. In our case, in Line 4, the solution x satisfies $|F|$ many linearly independent inequalities. Since the claim is about the case when we reach Line 6, none of these $|F|$ inequalities are of the form "$x_e \geq 0$". Therefore, there must exist $\geq |F|$ vertices v with $\sum_{e:v\in e} x_e = 1$.

  Indeed, let T be the subset of *tight* vertices as described above; so $|T| \geq |F|$. For $v \in T$, let $\deg(v)$ denote the number of edges of F incident on v. Now note that

$$\sum_{v\in T} \deg(v) \underbrace{=}_{\text{hypergraph handshake}} \sum_{e\in F} |e \cap T| \leq 3|F|$$

  Since $|T| \geq |F|$, we can assert that there must exist some $v \in T$ with $\deg(v) \leq 3$.

  Now consider the edge $e^* \in F$ with $v \in e^*$ and $x(e^*)$ largest among all such edges. We claim that this edge e suffices for the claim. Indeed, first note that

$$1 \underbrace{=}_{v\in T} \sum_{e:v\in e} x(e) \leq x(e^*) \cdot \deg(v) \leq 3x(e^*) \Rightarrow x(e^*) \geq \frac{1}{3}$$

  Now suppose $e^* = (v, w, z)$. Note that

$$x(N(e^*)) \leq \sum_{f:v\in f} x_f + \sum_{f:w\in f} x_f + \sum_{f:z\in f} x_f - 2x(e^*) \leq 3 - \frac{2}{3} = \frac{7}{3}$$

  where we subtracted $2x(e^*)$ because of double-counting. Indeed, there could be many edges which could be double counted, and that is why we have a inequality. This completes the proof of the claim.

Remark: *The above rounding algorithm is an **iterative rounding** algorithm. As stated above the algorithm seems inefficient as it solves an LP in Line 4. In reality, Line 4 and Line 5 can be taken outside the while loop. The reason is that after a run of the while loop, the "residual solution" is also a basic feasible solution to a slightly modified LP. We leave the details from these notes.*

Exercise: ☕☕ *Find a 3-uniform hypergraph H with $\mathsf{lp}(H) = \frac{7}{3} \cdot \mathtt{opt}(H)$ thus proving that the integrality gap of the LP(1) is exactly $\frac{3}{7}$.*

Notes

It is easy to generalize the above to obtain a $\left(k - 1 + \frac{1}{k}\right)^{-1}$-approximation algorithm for k-uniform hypergraph matching, where every hyperedge has exactly k-vertices. This result was first proved in the paper [4] by Füredi, Kahn and Seymour. In fact, [4] proved a more general result for an arbitrary hypergraph : they proved that in any hypergraph one can find a matching M such that $\sum_{e\in M} \left(|e| - 1 + \frac{1}{|e|}\right) \geq \mathsf{lp}$. The analysis above is from the paper [3] by Chan and Lau who also give an $\left(k - 1 + \frac{1}{k}\right)^{-1}$-approximation algorithm which works for the weighted case as well. We refer the reader to that paper for more details.

Füredi, Kahn, and Seymour [4] conjectured a weighted generalization of their theorem: they conjecture that with any weights $w(e)$ on edges, there exists a matching M such that $\sum_{e\in M} \left(|e| - 1 + \frac{1}{|e|}\right) w(e) \geq \mathsf{lp}$, where lp now has $w(e)$ in the objective of (1). This conjecture is still open. Very recently, it was proved for hypergraphs with all $|e| \leq 3$ in the paper [2] by Bansal and Harris. See also the recent paper [1] by Anegg, Angelidakis, and Zenklusen for more on the FKS conjecture.

References

[1] G. Anegg, H. Angelidakis, and R. Zenklusen. Simpler and stronger approaches for non-uniform hypergraph matching and the füredi, kahn, and seymour conjecture. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 196–203, 2021.

[2] N. Bansal and D. G. Harris. Some remarks on hypergraph matching and the Füredi-Kahn-Seymour conjecture. *arXiv preprint arXiv:2011.07097*, 2020.

[3] Y. H. Chan and L. C. Lau. On linear and semidefinite programming relaxations for hypergraph matching. *Math. Programming*, 135(1):123–148, 2012.

[4] Z. Füredi, J. Kahn, and P. D. Seymour. On the fractional matching polytope of a hypergraph. *Combinatorica*, 13(2):167–180, 1993.