

Lecture 20

Lecturer: Sofya Raskhodnikova

Scribe(s): Youngtae Youn

We continue the topic of testing monotonicity on general poset domains [1]. We first define what an induced matching is and explain what it looks like in a hypercube. Then we show that a graph with many *large* induced matchings is hard to test monotonicity. Finally, we explain how to construct such a graph to prove the lower bound of query complexity.

1 Induced Matching

A *matching* in a graph is a set of edges without any common vertices. An *induced matching* M of a graph $G = (V, E)$ is a set of edges $M \subseteq E$ such that M is a matching and no two edges of M are joined by an edge of G . It can be explained by the subgraph induced by $V(M)$, the endpoints of a matching M ; that is, the set of edges induced by $V(M)$ is the same as M . For example, consider a graph in Figure 1a with eight vertices and four edges. Picking $M = \{(b, e), (d, g)\}$ results in an induced matching in Figure

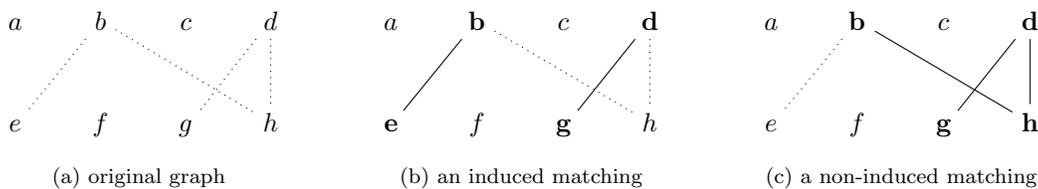


Figure 1: Example of an induced matching and a non-induced matching

1b as the subgraph induced by the set of vertices $\{b, d, e, g\}$ has only two edges (b, e) and (d, g) . On the contrary, picking $M = \{(b, h), (d, g)\}$ results in a non-induced matching in Figure 1c as the subgraph induced by $\{b, d, g, h\}$ has an extra edge (d, h) other than (b, h) and (g, d) .

An Induced Matching on \mathcal{H}_d . Consider an induced matching on a d -dimensional hypercube \mathcal{H}_d . A node in \mathcal{H}_d is a d -bit string and whose weight is the number of 1's in the string. One can treat a hypercube as a bipartite graph by putting even weight nodes on the bottom and odd weight nodes on the top. Observe that there are no edges between nodes in the same side. An induced matching in \mathcal{H}_d can be formed for every dimension $i \in [d]$ as follows:

1. Pick every bottom nodes with the i^{th} bit being 0.
2. For each bottom node $\alpha 0 \beta$, choose an edge $(\alpha 0 \beta, \alpha 1 \beta)$.

Figure 2 shows how to construct an induced matching in \mathcal{H}_3 by setting $i = 2$. Note that this construction always outputs an induced matching. Consider any two edges $(\alpha 0 \beta, \alpha 1 \beta)$ and $(\alpha' 0 \beta', \alpha' 1 \beta')$ returned by this construction. As $\alpha 1 \beta$ and $\alpha' 0 \beta'$ differs more than one bit, these two nodes are not connected by any edge. The same reasoning applies to $\alpha' 1 \beta'$ and $\alpha 0 \beta$ so that these two are not connected either.

Both the bottom and the top side in \mathcal{H}_d has 2^{d-1} nodes, respectively. And the half of the bottom nodes have the i^{th} bit 0. It means each such matching has $n/4 = 2^{d-2}$ edges. As our induced matching construction is determined by the choice of $i \in [d]$, there are $d = \log n$ such matchings.

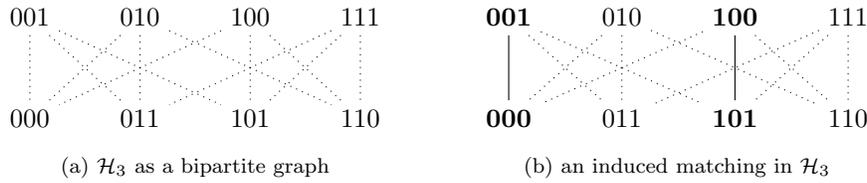


Figure 2: An induced matching in \mathcal{H}_3 obtained by setting $i = 2$.

2 Query Lower Bounds on General Bipartite Graphs

We just explained how to construct an induced matching in a hypercube. From now on, we switch our attention to general bipartite graphs with n vertices. Assume that all edges in these bipartite graphs go from a bottom node to a top node. Let m denote the number of induced matching in these graphs. An induced matching is *large* if its size is ϵn for some constant $\epsilon < 1$. Our goal is to prove the following.

Theorem 1. *A bipartite graph G with m large induced matchings requires $\Omega(\sqrt{m})$ queries for non-adaptive monotonicity testing.*

We will use the variant of Yao's principle introduced in the last lecture to prove the query complexity lower bound. Namely, we construct distributions \mathfrak{P} on positive (monotone) inputs and \mathfrak{N} on negative (ϵ -far from monotone) inputs such that it is hard to distinguish them with $q = o(\sqrt{m})$ queries. The construction is as follows:

For both \mathfrak{P} and \mathfrak{N} ,

1. Randomly pick one of m induced matching.
2. For non-matching edges (u, v) , set the top node v to 1 and the bottom node u to 0.
3. For each top node v in matching edges (u, v) , set it to either 0 or 1 with probability $1/2$.
4. For each bottom node u in matching edges (u, v) ,¹
 - (a) set $f(u) = f(v)$ for \mathfrak{P} .
 - (b) set $f(u) = 1 - f(v)$ for \mathfrak{N} .

This construction is correct, which means \mathfrak{P} has a monotone node labeling of the graph G and \mathfrak{N} has a node labeling that are ϵ -far from monotone.

Theorem 2. *The distribution \mathfrak{P} has a monotone node labeling of G .*

Proof. Consider two kinds of edges in G .

1. Edges in a matching: Two endpoints of such an edge are set to be equal by the last step of the construction.
2. Edges not in a matching: Bottom nodes are always set to 0 and top nodes to 1.

As all m matchings are induced matchings, G has these two kinds of edges only. None of the edges are violated, so the node labeling is monotone. \square

Theorem 3. *If the size of the matchings are at least $4\epsilon n$, then \mathfrak{N} has a labeling ϵ -far from monotone with overwhelming probability.*

¹One might be tempted to set $f(u) = 0$ for \mathfrak{P} and $f(u) = 1$ for \mathfrak{N} . Unfortunately, however, these distributions are easy to distinguish for any tester.

Proof. Consider any labeling in \mathfrak{N} . The probability that any edge from the selected matching is violated is $1/2$, which corresponds to setting the top node $f(v) = 0$ and the bottom node $f(u) = 1 - f(v)$ in our construction. Then the expected number of violated edges in this matching is at least $\frac{1}{2}4\epsilon n = 2\epsilon n$.

The probability that the node labeling is not ϵ -far from monotone is equal to the probability that the number of violated edges in the mapping is less than ϵn , which amounts to $\exp\{-\Omega(\epsilon n)\}$ by Chernoff bound. Hence, the node labeling is ϵ -far from monotone with probability at least $1 - \exp\{-\Omega(\epsilon n)\}$. \square

Proof of Theorem 1. Fix any non-adaptive deterministic tester with $q = o(\sqrt{m})$ queries. Let x_1, \dots, x_q denote the nodes that are queried by the tester. Let E be the events that the tester queries both endpoints in the selected matching. Note that all inputs from \mathfrak{P} and \mathfrak{N} were constructed in pairs based on a selected matching. As the top endpoints in this matching is labeled with a random binary value, an algorithm cannot tell the difference unless querying a matching edge. Hence, $(P - \text{view}|\overline{E}) = (N - \text{view}|\overline{E})$ where $(P - \text{view}|\overline{E})$ is P -view conditioned on \overline{E} and $(N - \text{view}|\overline{E})$ is N -view conditioned on \overline{E} .

Meanwhile, the probability of E is bounded by:

$$\Pr\{E\} \leq \sum_{(i,j) \in \{x_1, \dots, x_q\}} \Pr\{(i,j) \in \text{selected matching}\} \leq q^2/m = o(1)$$

where the second inequality follows from the fact that any edge belongs to a unique matching with probability at most $1/m$. It leads to the conclusion that $SD(P - \text{view}, N - \text{view}) = o(1)$. Combined with Yao's principle, the query complexity of non-adaptive 1-sided error testers must be $\Omega(\sqrt{m})$. \square

3 Constructing Hard-to-Test Graphs

To achieve our lower bound, we show that there is a graph with n vertices that can be partitioned into m matchings of size s where s is linear in n . Ruzsa and Szemerédi [2] constructed such a graph with $t = n/3$, but we cannot use their construction since the matching size s is *nearly* linear in n . In the next lecture, we construct such a graph with the matching size $s = \frac{n}{8} - o(n)$ and the number of matchings $m = n^{\Omega(1/\log \log n)}$.

References

- [1] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. *Monotonicity testing over general poset domains*, STOC 2002.
- [2] Imre Ruzsa and Endre Szemerédi. *Triple systems with no six points carrying three triangles*, Combinatorics (Keszthely, 1976), Coll. Math. Soc. J. Bolyai, 1978