Polyhedral combinatorics – Lecture 12

Cyril Kotecký

14 June, 2024

version 1 - June 16, 2024

Definition: ***Randomized protocol***

- A, B finite sets assigned to Alice and Bob respectively

A **randomized protocol** is a rooted binary tree such that:

1. Each node of the tree has type A or B
2. Each node v of type A has functions $p_{0,v}, p_{1,v} : A \to [0,1]$ such that $p_{0,v}(i) + p_{1,v}(i) \leq 1$.
3. Each node v of type B has functions $q_{0,v}, q_{1,v} : B \to [0,1]$ such that $q_{0,v}(j) + q_{1,v}(j) \leq 1$.
4. Each leaf v of type A has a nonnegative vector Λ_v of size $|A|$.
5. Each leaf v of type B has a nonnegative vector Λ_v of size $|B|$.

- Nowhere is it mentioned that vertices of type A have vertices of type B as children and vice versa, but this seems implicit from the context.

An **execution** of the protocol on input $(i, j) \in A \times B$ is a random path from the root.

- It descends to the left child of an internal node v with probability $\begin{cases} p_{0,v}(i) & v \text{ is of type } A \\ q_{0,v}(j) & v \text{ is of type } B \end{cases}$ and to its right child with probability $\begin{cases} p_{1,v}(i) & v \text{ is of type } A \\ q_{1,v}(j) & v \text{ is of type } B \end{cases}$.
- The execution stops at v with probability $\begin{cases} 1 - p_{0,v}(i) - p_{1,v}(j) & v \text{ is of type } A \\ 1 - q_{0,v}(i) - q_{1,v}(j) & v \text{ is of type } B \end{cases}$.
- If an execution stops at a node v, the **value** of the execution is $\begin{cases} 0 & v \text{ is an internal node} \\ \Lambda_v(i) & v \text{ is a leaf of type } A \\ \Lambda_v(j) & v \text{ is a leaf of type } B \end{cases}$.
- For a fixed input $(i, j) \in A \times B$, the value of the execution is a random variable.
- Transitioning from a node A to the left or right child corresponds to Alice sending 0 or 1 respectively, and symmetrically for B.

The **communication complexity** of the protocol is the maximum number of bits exchanged over all (i, j), or equivalently, the height of the tree.

Problem: ***Computing a matrix in expectation***

- For a given matrix M and a protocol outputting X, the goal is to get $\mathbb{E}[X_{i,j}] = M_{i,j}$ for all entries.
- The **communication complexity** $cc_+(M)$ of the protocol is the maximum number of bits exchanged.

Theorem:

$$\lceil \log \mathrm{rk}_+(M) \rceil = \mathrm{cc}_+(M)$$

Proof.

$\leq$
- Assume we have a protocol computing X with $\mathbb{E}[X_{xy}] = M_{xy}$ with complexity c.
- Each node v of the protocol has a corresponding traversal probability matrix $P_v \in \mathbb{R}_+^{A \times B}$ with $\forall (x, y) \in A \times B : \ P_v(x, y) = P[\text{execution on input } (x, y) \text{ goes through } v]$. Let $v_1, ..., v_k$ be the nodes of type A on the unique path P from the root r to the parent of v. Let $w_1, ..., w_l$ be the nodes of type Y on this path. Then

$$P_v(x,y) = \prod_{i\in[k]} p_{\alpha_i,v_i}(x) \prod_{j\in[l]} q_{\beta_j,w_j}(y),$$

where $\alpha_i = \begin{cases} 1 & \text{path goes into right subtree of } v_i \\ 0 & \text{path goes into left subtree of } v_i \end{cases}$ and similarly for β_j. Hence P_v is a rank one matrix of the form $a_v b_v$ for a_v column vector of size $|A|$ and b_v row vector of size $|B|$.

- Let L_X, L_Y be the sets of leaves of types A and B respectively. Let Λ_v denote the vector of values at a leaf $v \in L_X \cup L_Y$. Since the protocol computes $\mathbb{E}[X_{xy}] = M_{xy}$, we have

$$M(x,y) = \sum_{v\in L_X} \Lambda_v(x) P_v(x,y) + \sum_{w\in L_y} P_w(x,y)\Lambda_w(y).$$

Thus

$$M = \sum_{v\in L_X} (\Lambda_v \circ a_v) b_v + \sum_{w\in L_Y} a_w (b_w \circ \Lambda_w)$$

where $\circ$ denotes the hadamard (element-wise) product. Hence we can express M as a sum of at most $|L_X \cup L_Y| \leq 2^c$ nonnegative rank one matrices and therefore $rank_+(M) \leq 2^c$.

$\geq$

- Denote $r := \mathrm{rk}_+(M)$ and let $A \in \mathbb{R}_+^{m\times r},\ B \in \mathbb{R}_+^{r\times n}$ be such that $M = AB$. WLOG we can assume that the maximum row sum of A is 1, as we can rescale B appropriately.
- Alice knows a row index i and Bob knows a column index j. Together, they want to compute $\mathbb{E}[X_{ij}] = M_{ij}$ by exchanging as few bits as possible.

1. Alice selects a column index $k \in [r]$ according to the probabilities in row i of A. She sends this index to Bob.
2. Bob outputs entry of B_{kj}.

- With probability $1 - \delta_i$, where $\delta_i := \sum_k A_{ik} \leq 1$, Alice does not send any index to Bob and the computation stops with implicit output zero.
- This randomized protocol computes X with $\mathbb{E}[X_{ij}] = M_{ij}$, since for the input (i,j), the expected value is $\sum_{k\in[r]} A_{ik}B_{kj} = M_{ij}$. Moreover, the complexity of the protocol is precisely $\lceil \log r \rceil$.

□

Corollary:

- $P \neq \emptyset$ polytope that is not a point
- S its associated slack matrix

$$\lceil \log \mathrm{xc}(P) \rceil = \mathrm{cc}_+(S)$$

Proof.

- Follows from the previous theorem and Yannakis' theorem.

□

1 Perfect matching polytope

Definition: ***Perfect matching polytope***

$$P_{PM} := \mathrm{conv}\{\chi^{E'} \mid E' \subseteq E \text{ perfect matching}\}$$

$$= \{x \in \mathbb{R}^{|E|} \mid \forall v \in V : \sum_{v\in e} x_e = 1,\ x_e \geq 0,\ \forall U \in V,\ |U| \text{ odd: } \sum_{e\in\delta(U)} x_e \geq 1\}$$

$$\text{or } \{x \in \mathbb{R}^{|E|} \mid \forall v \in V : \sum_{v\in e} x_e = 1,\ x_e \geq 0\} \text{ for bipartite graphs.}$$

Note: ***Perfect matching polytope***

- Complexity is exponential.
- Looking at a vertex and its neighbors, we have a polyhedral cone called the **vertex figure**.
- Vertex figures of perfect matching polytope have small extension complexity.