The complexity of matrix rank and feasible systems of linear equations

Eric Allender   Robert Beals   Mitsunori Ogihara

Definitions

1. For a NDTM M, define the function gap_M as $\text{gap}_M(x) = \#\text{accepting paths of } M \text{ on } x - \#\text{rejecting paths of } M \text{ on } x$.
2. $\text{GapL} = \{\text{gap}_M |$ M is a NL machine$\}$.
3. $\text{C}_=\text{L} = \{A | \exists f \in \text{GapL}$ such that $x \in A \iff f(x) = 0\}$.
4. A problem Q is logspace-uniform NC^1-reducible to F i.e. $Q \in NC^1(F)$ if there exists a logspace machine M such that for every $n \geq 1$, M on 1^n outputs a description of a circuit $C_n \in NC^1$ which can contain oracle gates from F and for every x of length n, C_n outputs $Q(x)$ on input x.

Results in the paper

Theorem (1)

The sets $\{(M, r) |\ M \in \mathbb{Z}^{n \times n}$ *and* $rank(M) = r\}$ *and* $\{M |\ M \in \mathbb{Z}^{n \times n}$ *and* $rank(M) = n - 1\}$ *are complete for* $C_{=}L \cap$ *co-*$C_{=}L$.

Theorem (2)

$L^{C_{=}L} = NC^1(C_{=}L)$.

Review of some previously known results

[Ber84] **Computing coefficients of characteristic polynomial** $\leq_L$ **Iterated matrix multiplication**

$\implies$ **Determinant** $\leq_L$ **Iterated matrix multiplication** [$\because$ Constant term of characteristic polynomial of A is $(-1)^n \det(A)$]

[Val92] Iterated matrix multiplication $\leq_L$ Determinant

There is a logspace-computable function that takes as input a sequence of matrices D_i and numbers (a, b) and outputs a matrix H such that entry (a, b) of $\prod D_i$ is $\det(H)$.

- Build a layered directed graph where edges exist only between adjacent layers. The edge between k-th vertex of i-th layer and m-th vertex of $(i+1)$-th layer will have weight equal to entry (k, m) of D_i. Then entry (a, b) of $\prod D_i$ equals to the sum of weights of all paths from vertex a in the first layer to vertex b in the last layer [weight of a path is the product of weights of the edges on the path].
- Replace every edge (x, y) of weight c with a path of length 2 consisting of an edge (x, z) having weight 1 and an edge (z, y) having weight c. This make every path from the first layer to the last layer of even length.
- Add self-loops of weight 1 to all vertices except vertex b in the last layer.
- Add an edge from vertex b in the last layer to vertex a in the first layer of weight 1.
- If H is the adjacency matrix of this graph, then $\det(H)$ has the desired value.

Corollary

There is a logspace-computable function f such that if M is a matrix of full rank, then so is $f(M)$, and if $\det(M) = 0$, then $f(M)$ has rank exactly one less than full.

- Using **Determinant $\leq_L$ Iterated matrix multiplication**, we obtain matrices D_i such that $(1, n)$ entry of $\prod D_i$ is $\det(M)$.
- Then we use the last lemma to obtain $H_{n\times n}$ such that $\det(H) = \det(M)$.
- For $1 \leq i, j \leq n - 1$, (i, j) is an edge in the graph induced by $H \implies i \leq j$. Hence the submatrix of H induced by the first $(n - 1)$ rows and the first $(n - 1)$ columns is upper triangular. As the diagonal entries of H are all 1, we have that $\text{rank}(H) \geq n - 1$.

[Mul87] **Checking whether the rank of an $n \times n$ matrix A is $\leq k$**

- Take $A' = \begin{bmatrix} 0 & A \\ A^t & 0 \end{bmatrix}$. We have $\mathrm{rank}(A') = 2 \cdot \mathrm{rank}(A)$.
- Let Y be a $2n \times 2n$ diagonal matrix with $Y_{ii} = y^{i-1}$ for $1 \leq i \leq 2n$ for some indeterminate y. Take $B = YA'$. Then $\mathrm{rank}(B) = \mathrm{rank}(A')$.
- $\mathrm{Ker}(B^2) = \mathrm{Ker}(B) \implies \mathrm{Ker}(B^t) = \mathrm{Ker}(B)\ \forall t \geq 1 \implies \cup_{t \geq 1} \mathrm{Ker}(B^t) = \mathrm{Ker}(B)$.
  Fact from linear algebra: $\dim(\cup_{t \geq 1} \mathrm{Ker}(B^t))$ equals to the highest integer m such that t^m divides the characteristic polynomial $P(t)$ of B.
  Then $m = \dim(\mathrm{Ker}(B)) = 2n - \mathrm{rank}(B)$.

- The first $2n - \text{rank}(B)$ coefficients of $P(t)$ are all zero. We can use **Computing coefficients of characteristic polynomial $\leq_L$ Iterated matrix multiplication** to build matrices D_i such that it suffices to check whether certain entries of $\prod D_i$ are zero. The entries of D_i are polynomials of degree at most $4n^2$.
- For any matrix $X \in K[y]^{m \times m}$, write $X = X_0 + X_1 y + X_2 y^2 + \dots$

  The map $\phi_d : K[y]^{m \times m} \to (K^{d \times d})^{m \times m}$ defined by $\phi_d(X) = \begin{bmatrix} X_0 & X_1 & \dots & X_{d-1} \\ 0 & X_0 & \dots & X_{d-2} \\ . & . & . & . \\ 0 & 0 & \dots & X_0 \end{bmatrix}$ is a ring homomorphism. Thus it suffices to check whether certain entries of $\prod \phi_d(D_i)$ are zero.

Complete problem for $C_=L$

The set $A = \{(M, r) | M \in \mathbb{Z}^{n\times n}$ and $\text{rank}(M) < r\}$ is complete for $C_=L$.

Hardness: Determinant is GapL-hard $\implies$ Singularity is $C_=L$-hard $\implies$ A is $C_=L$-hard.
Inclusion in $C_=L$:

- Iterated integer matrix multiplication $\in$ GapL $\implies$ Given integer matrices D_i and integers (a, b), the problem of checking whether an entry (a, b) of $\prod D_i$ is zero is in $C_=L$.
- The preceding discussion shows that the problem of determining if the rank is $\leq r - 1$ is logspace conjunctive-truth-table reducible to a problem in $C_=L$ [f is a logspace conjunctive truth-table reduction from A to B if for all x, $x \in A \iff f(x, i) \in B$ for all $i \leq poly(|x|)$].
- [AO96] $C_=L$ is closed under logspace conjunctive-truth-table reductions.

Complete problem for $\mathrm{C_=L} \cap \text{co-}\mathrm{C_=L}$

The sets $\{(M, r) \mid M \in \mathbb{Z}^{n \times n} \text{ and } \operatorname{rank}(M) = r\}$ and $\{M \mid M \in \mathbb{Z}^{n \times n} \text{ and } \operatorname{rank}(M) = n - 1\}$ are complete for $\mathrm{C_=L} \cap \text{co-}\mathrm{C_=L}$.

Let $A = B \cap C$ where $A \in \mathrm{C_=L}$ and $B \in \text{co-}\mathrm{C_=L}$.

- Since the set of singular matrices is complete for $\mathrm{C_=L}$, we can compute matrices M_1 and M_2 such that $x \in A \iff \det(M_1) = 0$ and $\det(M_2) \neq 0$.
- We can compute matrices M_3 and M_4 such that $x \in A \iff$ rank of M_3 is one less than full and rank of M_4 is full.
- $x \in A \iff$ rank of $\begin{bmatrix} M_3 & & \\ & M_4 & \\ & & M_4 \end{bmatrix}$ is one less than full.

Theorem

$L^{C_=L} = NC^1(C_=L)$.

$L^{\mathrm{C_=L}} \subseteq AC^0(\mathrm{C_=L}) \subseteq NC^1(\mathrm{C_=L})$.

$NC^1(\mathrm{C_=L}) \subseteq L^{\mathrm{C_=L}}$:

Let B be a language in $NC^1(\mathrm{C_=L})$ and $\{C_n\}_{n\geq 1}$ be a logspace-uniform NC^1 circuit family that reduces B to $A \in \text{co-C}_=\text{L}$. Let N be a NDTM witnessing that $A \in \text{co-C}_=\text{L}$.

Assumptions:

- N has a one-way input tape.
- Each C_n is a tree.
- Each gate of C_n is either an input gate or an oracle gate.

For each oracle gate g in C_n, we assign weight $R(g) = 2^k$ where k is the number of oracle gates in C_n between g and the output gate.

Define a machine M as follows:

On input (x, m), M guesses in the following manner a collection H of nodes in C_n such that the sum of their weights is equal to m:

1. First M initializes a variable s to m. Then it traverses C_n by depth-first search. Whenever it visits a new node g, it guesses the output of that gate.
2. If the guessed output of g is 1, then M subtracts $R(g)$ from s and simulates N on the input of g.
3. If the guessed output of g is 0, M continues traversing the tree.
4. If g is an input gate and the guessed output of g doesn't match with its actual value, then abort the simulation and create an accepting and a rejecting path.
5. At the end, if $s \neq 0$, then M creates an accepting path and a rejecting path. Otherwise, M accepts if and only if the number of simulations of N where a rejecting state is encountered is even.

Fix H. Let $g_1, \ldots, g_m$ be the gates in H and $y_1, \ldots, y_m$ be their inputs. Then the gap generated by M is $\mathrm{gap}_N(y_1) \ldots \mathrm{gap}_N(y_m)$.

Let Z_x be the actual collection of oracle gates of C_n that output 1 on input x and let n_x be the sum of the weights of all gates in Z_x.

Want to show: n_x is the largest m such that M generates non-zero gap on input (x, m).

- If M correctly guesses Z_x as H, then gap generated is non-zero since $y_i \in A$ for all i.
- Consider $Z \neq Z_x$ such that the weight sum of Z is $\geq n_x$.

  The weight of any gate is greater than the sum of weights of all of its ancestors. Hence there is a gate g is $Z \backslash Z_x$ such that for all gates h below g,

$$h \in Z_x \iff h \in Z \ .$$

  When M guesses Z as H, the input string for g is the same as what it would be if M would have guessed Z_x as H (i.e. the actual query string for g).

  Hence, $\mathrm{gap}_N(u) = 0$ implying that the gap generated by M is zero.

Let M' be the machine that does everything the same as M' except that it guesses the output of the output gate to be 1.

Let X_1, X_2 be languages in co-$\mathrm{C_{=}L}$ defined by the gap function of M and M' respectively.

Then $x \in B \iff$ The highest $m \leq q(|x|)$ for which $(x, m) \in X_1$ also satisfies $(x, m) \in X_2$.

$\therefore B \in L^{\mathrm{C_{=}L}}$.