Lecture 1 (27.02.2013)

Scribe: Łukasz Solak
Lecturer: Marek Cygan

Algebraic methods

1 Introduction

Presenting problems in terms of algebra leads to simple algorithms, in order to solve a problem sometimes it is enough to check if determinant or rank of a specific matrix satisfies a simple condition.

We will often use matrix multiplication in algorithms thus time in which we can multiply matrices is crucial for analysis of algorithms. Because new algorithms for this problems appear, we will denote time complexity of multiplying two $n \times n$ matrices over some ring as $\mathcal{O}\left(n^{\omega}\right)$ where ω stands for the lowest known exponent which currently is less than 2.3727 [3]. An example of algorithm that multiplies matrices faster than $\mathcal{O}\left(n^3\right)$ is Strassen's algorithm [2]. Other algorithms for matrix multiplication problem which are faster than Strassen's algorithm asymptotically aren't practical because of high constant factor hidden in $\mathcal{O}$ notation.

Remark 1. Time complexity of algorithm for matrix multiplication might differ depending on chosen computation model. For example if we choose $\mathbb{Z}_m$ with large m for a ring then it is reasonable to say that it contributes as $\mathcal{O}(\log_2(m))$ factor to algorithm's time complexity. To deal with this we introduce $\tilde{\mathcal{O}}$ notation that omits logarithmic factors.

Lemma 2. *If A is a square matrix of size n over some field, then it is possible to find* $\det(A)$, $\operatorname{rank}(A)$ *and* A^{-1} *in* $\tilde{\mathcal{O}}\left(n^{\omega}\right)$ *time.*

In this lecture we extensively use computing matrix's determinant as a subproblem and the following equality comes handy in proofs:

$$\det(A) = \sum_{\pi \in S_n} \operatorname{sgn}(\pi) \prod_{1 \le i \le n} a_{i,\pi(i)}$$

where S_n is set of permutations of n-set (n being the size of the square matrix A).

2 Perfect matching in balanced bipartite graphs

We study the following problem.

Perfect matching in balanced bipartite graphs
Input: Bipartite graph G = (U ∪ V, E) where $|U| = |V| = n$
Question: Does there exists a perfect matching in G?

Let's define a matrix $A(G)$ where $a_{u,v} = \begin{cases} x_{u,v} & \text{if } (u,v) \in E \\ 0 & \text{otherwise} \end{cases}$

Theorem 3 (Edmonds)**.** *There exists a perfect matching in a balanced bipartite graph G if and only if* $\det(A(G)) \neq 0$.

Proof. ($\Rightarrow$) If $\det(A(G)) \neq 0$, then there must exist $\delta \in S_n$ such that $\operatorname{sgn}(\delta)\prod_{1\leq i\leq n} a_{i,\delta(i)} \neq 0$ therefore there exists a perfect matching consisting of edges $(u, \pi(u))$ where $u \in U$.

($\Leftarrow$) Suppose there is a perfect matching, then there exists $\delta \in S_n$ such that $u \in U$ is connected with $\delta(u)$ in this matching. Therefore it is evident that $\operatorname{sgn}(\delta)\prod_{1\leq i\leq n} a_{i,\delta(i)} \neq 0$ moreover for $\pi_1, \pi_2 \in S_n$ such that $\pi_1 \neq \pi_2$ if $\prod_{1\leq i\leq n} a_{i,\pi_1(i)} \neq 0 \wedge \prod_{1\leq i\leq n} a_{i,\pi_2(i)} \neq 0$, then $\prod_{1\leq i\leq n} a_{i,\pi_1(i)} \neq \prod_{1\leq i\leq n} a_{i,\pi_2(i)}$ because of variables mismatch thus $\det(A(G)) \neq 0$. □

Remark 4. Note that symbolic determinant might grow large therefore we need some other way to check if it is nonzero. One possibility is to use Lovasz's algorithm described later.

Lemma 5 (Schwartz-Zippel). *Let $P \in F[x_1, x_2, \ldots, x_n]$ be a nonzero polynomial of total degree $d \geq 0$ over a field, F. Let S be a finite subset of F and let $r_1, r_2, \ldots, r_n$ be selected randomly from S. Then $Pr[P(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d}{|S|}$.*

Proof. The proof is by mathematical induction on the number of variables. For $n = 1$ this follows directly from the fact that a polynomial of degree d can have no more than d roots. Suppose the theorem holds for all polynomials in $n - 1$ variables. We can consider P to be a polynomial in x_1 by writing it as $P(x_1, \ldots, x_n) = \sum_{i=0}^{d} x_1^i P_i(x_2, \ldots, x_n)$.

Since P is not identically 0, there is some i such that P_i is not identically 0. Let's denote the largest such i as i_0. Then $\deg P_{i_0} \leq d - i_0$.

Now we randomly pick $r_2, \ldots, r_n$ from S. By the induction hypothesis, $Pr[P_{i_0}(r_2, \ldots, r_n) = 0] \leq \frac{d-i_0}{|S|}$. If $P_{i_0}(r_2, \ldots, r_n) \neq 0$, then $P(x_1, r_2, \ldots, r_n)$ is of degree i_0 so $Pr[P(r_1, r_2, \ldots, r_n) = 0 | P_{i_0}(r_2, \ldots, r_n) \neq 0] \leq \frac{i_0}{|S|}$

If we denote the event $P(r_1, r_2, \ldots, r_n) = 0$ by A, the event $P_{i_0}(r_2, \ldots, r_n) = 0$ by B we have $Pr[A] = Pr[A \cap B] + Pr[A \cap B'] = Pr[B]Pr[A|B] + Pr[B']Pr[A|B'] \leq Pr[B] + Pr[A|B'] \leq \frac{d-i_0}{|S|} + \frac{i_0}{|S|} = \frac{d}{|S|}$ □

Please recall that it is possible to compute the determinant of a square matrix over a field in $\tilde{\mathcal{O}}(n^\omega)$ time.

Algorithm 6 (Lovasz). *To test whether* $\det(A) = 0$ *for the matrix A (with elements being polynomials) choose some field F (for example Z_p for some large prime p) and randomly assign values to variables, and then check whether the determinant of a resulting matrix (with elements from F) is equal to zero.*

Remark 7. Using the Schwartz-Zippel lemma we can estimate that the probability of Lovasz's algorithm saying that the determinant is equal to zero when it is not (it is always correct the other way around) is lower than $\frac{1}{n}$ when we choose F such that $|F| = \Omega(n^2)$. The probability of a false zero can be reduced further by repeating the algorithm or choosing a field with more elements.

Corollary 8. *Given a bipartite graph G it is possible to check whether there exists a perfect matching in time $\tilde{\mathcal{O}}(n^\omega)$ with high probability.*

3 Perfect matching in arbitrary graphs

PERFECT MATCHING IN ARBITRARY GRAPHS
Input: Graph $G = (V, E)$
Question: Does there exists a perfect matching in G?

Definition 9 (Tutte's matrix)**.** Matrix $A(G)$ where $a_{u,v} = \begin{cases} x_{u,v} & \text{if } u < v \wedge (u,v) \in E \\ -x_{v,u} & \text{if } u > v \wedge (u,v) \in E \\ 0 & \text{otherwise} \end{cases}$

Theorem 10 (Tutte)**.** *There exists a perfect matching in G iff* $\det(A(G)) \neq 0$*.*

Proof. Let $H = \{\pi \in S_n :$ there exists at least one odd cycle in $\pi\}$ then we can write $\det(A(G)) = \sum_{\pi \in H} \operatorname{sgn}(\pi) \prod_{1 \leq i \leq n} a_{i,\pi(i)} + \sum_{\pi \in S_n \setminus H} \prod_{1 \leq i \leq n} a_{i,\pi(i)}$. Firstly let's prove that the first sum equals zero. We will do it by pairing permutations in H such that sum of each pair will be zero. Let $f(\pi)$ be a permutation that comes from π by reversing the odd cycle which contains the vertex with the lowest number. Note that f creates the mentioned pairing. For each $\pi \in H$, we pair π with $f(\pi)$. It creates pairing because $f(f(\pi)) = \pi$. if $\pi \in H$ then $\operatorname{sgn}(\pi) = \operatorname{sgn}(f(\pi))$ and $\prod_{1 \leq i \leq n} a_{i,f(\pi)(i)} = -\prod_{1 \leq i \leq n} a_{i,\pi(i)}$ because reversing the odd cycle provide odd number of sign changes therefore each pair sums up to zero.

If $\pi \in S_n \setminus H$ and $\prod_{1 \leq i \leq n} a_{i,\pi(i)} \neq 0$ then edges $(u, \pi(u)), u \in V$ create a directed cycle cover of G consisting of even length cycles. Note that two elements of the sum consist of the same variables (including multiplicities) iff the corresponding permutations have the same underlying undirected cycle covers.

($\Rightarrow$) Suppose there exists a perfect matching. We can create a directed cycle cover of G by taking each edge of the perfect matching two times, once for each direction. There's exactly one permutation π corresponding to the created cycle cover therefore $\det(A(G)) \neq 0$.

($\Leftarrow$) If $\det(A(G)) \neq 0$ then there exists $\delta \in S_n \setminus H$ such that $\prod_{1 \leq i \leq n} a_{i,\pi(i)} \neq 0$ therefore we can obtain a perfect matching by taking every second edge from each cycle of the cycle cover corresponding to δ. □

4 Finding the shortest cycle in directed graphs

4.1 Testing whether directed graph contains nontrivial cycle

First, we are going to solve the following problem, which is trivially solvable using graph searching algorithms, as it will be helpful in further sections. Note that in the following definition a cycle has to be of lenght at least two.

TESTING WHETHER DIRECTED GRAPH CONTAINS A CYCLE
Input: Directed graph $G = (V, E)$
Question: Does there exist a cycle in G?

Let's define a matrix $A(G)$ where $a_{u,v} = \begin{cases} 1 & \text{if } u = v \\ x_{u,v} & \text{if } (u,v) \in E \\ 0 & \text{otherwise} \end{cases}$

Lemma 11. *There exists a cycle in G iff* $\det(A(G)) \neq 1$*.*

Proof. Note that identity permutation contributes to the determinant's sum the value of 1.

$\Rightarrow$ Suppose there exists a cycle $v_1 - v_2 - \ldots - v_k - v_1$, $v_i \neq v_j$ for $i \neq j$ then let's take the permutation δ such that $\pi(v_i) = v_{i+1}$ for $1 \leq i < k$, $\pi(v_k) = v_1$ and identity for other vertices. Then $\prod_{1 \leq i \leq n} a_{i,\pi(i)} \neq 0$ and because every other nonzero element of the sum consists of different set of variables we conclude that $\det(A(G)) \neq 1$

$\Leftarrow$ If $\det(A(G)) \neq 1$ then there exists a permutation δ such that it is not identity and $\prod_{1 \leq i \leq n} a_{i,\delta(i)} \neq 0$ therefore there exists a cycle in the graph corresponding to the cycle in δ. □

4.2 Finding the shortest cycle in directed graphs

Finding the shortest cycle in directed graph without negative cycles
Input: Directed weighted graph G without a negative cycle
Question: Edges that create the shortest cycle in G

The algorithm consists of the following steps

1. Find the length of the shortest cycle
2. Find all edges that are part of some shortest cycle (one edge would suffice but we can find all without additional effort)
3. Find the shortest cycle

Let $l : E \to \{-M, \ldots, M\}$ be length function.

4.2.1 Step 1.

Let's define a matrix $A(G)$ where $a_{u,v} = \begin{cases} 1 & \text{if } u = v \\ y^{l(u,v)} \cdot x_{u,v} & \text{if } (u,v) \in E \\ 0 & \text{otherwise} \end{cases}$

For a multi-variable polynomial q, let us denote by:

- $\deg^*_y(q)$ – the degree of the smallest degree term of y in q
- $term^d_y(q)$ – the coefficient of y^d in q
- $term^*_y(q)$ – $term^d_y(q)$ for $d = deg^*_y(q)$

Note that $deg^*_y(det(A(G)) - 1)$ is the length of the shortest cycle.

Theorem 12 (Strojohann)**.** *The determinant of a matrix over single variable polynomials of degree $\leq d$ can be computed in $\tilde{\mathcal{O}}(dn^\omega)$.*

To apply Strojohann's theorem we must get rid of negative exponents, therefore we multiply every element of the matrix by y^M. So now we are looking for $\deg^*_y(\det(A(G) \cdot y^M) - y^{nM}) - nM$ which we can find in the following way.

1. Randomly assign values to variables of $A(G)$ (except for y) from some field F, $|F| = \mathcal{O}(n^2)$
2. Compute $\gamma = \det(A(G) \cdot y^M) - y^{nM}$
3. Return $\deg^*_y(\gamma) - nM$

4.2.2 Step 2.

Definition 13 (Straight-line program)**.** Given a set S, a straight-line program (SLP) is a family of functions $\mathcal{F} = \{f_i : 1 \leq i \leq m\}$, $f_i : S^{n+i-1} \to S$ for some fixed $n \in \mathbb{N}$. An SLP is evaluated on a tuple $(s_1, \ldots, s_n)$ by recursion in the following way: f_1 is evaluated on $(s_1, \ldots, s_n)$ as a function. The remaining evaluations are recursive, $\hat{f}_{i+1}(s_1, \ldots, s_n) = f_{i+1}(s_1, \ldots, s_n, f_1(s_1, \ldots, s_n), \ldots, f_i(s_1, \ldots, s_n))$. $\hat{f}_m(s_1, \ldots, s_n)$ is treated as a final output of SLP program.

The term straight-line reflects the fact that evaluating an SLP can be achieved by a program which does not branch or loop so its execution is a straight-line. It is common for SLPs to be built entirely from simple functions such as $f(x, y) = x + y$ or $f(x) = x \cdot x$.

Theorem 14 (Baur-Strassen [1])**.** *If an algorithm in SLP (straight-line program) form computes $f(x_1, \ldots, x_n)$ using T operations then using $3T$ operations it is possible to compute $\frac{\partial f}{\partial x_i}$ for every $i \in \{1, \ldots, n\}$ simultaneously.*

Lemma 15. *Strojohann's algorithm can be written in SLP form.*

Lemma 16. *$x_{u,v}$ lie on some shortest cycle iff $\frac{\partial\, \mathrm{term}^*_y(\det(A(G) \cdot y^M) - y^{nM})}{\partial x_{u,v}} \neq 0$*

Proof. Easy exercise for the reader. □

4.2.3 Step 3.

When we have an arc (u, v), that belongs to some shortest cycle it is enough to find the shortest path from v to u in the graph G. The algorithms computing shortest paths will be the subject of the next lecture.

References

[1] Jacques Morgenstern. How to compute fast a function and all its derivatives: a variation on the theorem of Baur-strassen. *SIGACT News*, 16(4):60–62, 1985.

[2] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969.

[3] Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In *STOC*, pages 887–898, 2012.