Complexity Theory of Polynomial-Time Problems

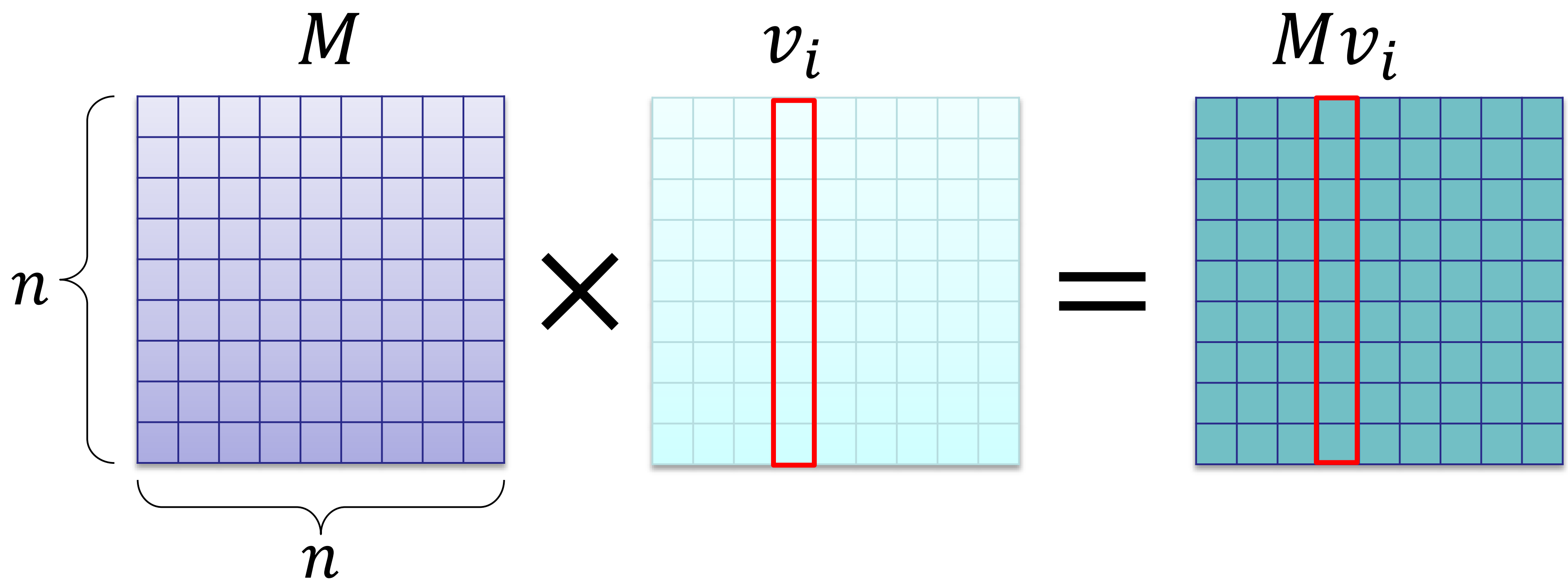
Lecture 12: More on OMv

Sebastian Krinninger
(Based on reading group talk of Pavel Kolev)

Online Boolean Matrix Multiplication

Input: Boolean $n \times n$ matrix M
Online sequence of vectors $v_1, \ldots, v_n \in \{0,1\}^n$

Output: Mv_i **before** v_{i+1} arrives *("query")*



OMv Conjecture: No algorithm with total time $O(n^{3-\epsilon})$ (for some $\epsilon > 0$).
(not even with polynomial-time preprocessing)

[Henzinger et al.'15]

A New Upper Bound

Theorem: OMv can be solved in total time $n^3/2^{\Omega(\sqrt{\log n})}$.

[Larsen, Williams'16]

Amortized time per query: $n^2/2^{\Omega(\sqrt{\log n})}$
for sequence of $\geq 2^{\Omega(\sqrt{\log n})}$ queries
(and no preprocessing time)

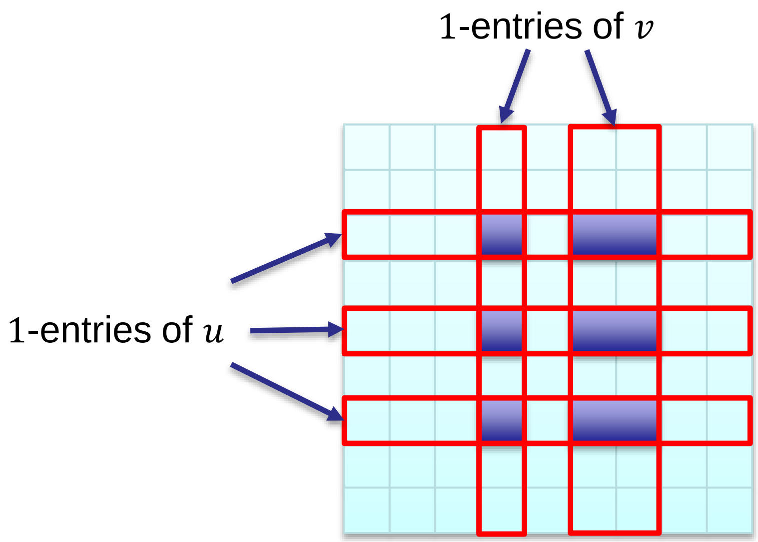
Lemma: If OuMv can be solved in total time $O(n^3/f(n))$, then OMv can be solved in total time $O(n^3/\sqrt{f(n)})$.

[Henzinger et al.'15]

This application: $f(n) = n^2/2^{\Omega(\sqrt{\log n})}$

OuMv: Vector-Matrix-Vector Multiplication

Observation: uMv iff submatrix of M induced by u and v contains a 1



Notation:

- $U \subseteq [n]$: set of indices with 1-entries in u
- $V \subseteq [n]$: set of indices with 1-entries in v
- $M[U \times V]$: submatrix of M induced by U and V

Data Structures

- $C \subseteq [n] \times [n]$
- List L of triples (U_k, V_k, S_k) s.t.
  - $U_k \subseteq [n]$
  - $V_k \subseteq [n]$
  - $S_k \subseteq [n] \times [n]$

Invariants:

- S_k contains all pairs (i, j) with $i \in U_k, j \in V_k$ s.t. $M[i, j] = 1$
  Intuition: (U_k, V_k) represents expensive query from the past
- C contains all pairs (i, j) that appear in no $U_k \times V_k$ of L
  Indicator matrix D s.t. $D[i, j] = 1$ iff $(i, j) \in C$
  Intuition: C contains unseen pairs

The Core Problem

At some point in algorithm: want to list all unseen pairs

Unseen Pairs: Given U and V, such that $|(U \times V) \cap C| \leq K$.
Determine $W := (U \times V) \cap C$

Idea: Reduce to listing orthogonal vectors:

C contains all pairs (i, j) that appear in no $U_k \times V_k$ of L

Define vectors $u_1, \dots, u_n, v_1, \dots, v_n$ of dimension $d := |L|$
For every $i \in [n]$: $u_i \in \{0,1\}^d$ s.t. $u_i[k] = 1$ iff $i \in U_k$
For every $j \in [n]$: $v_j \in \{0,1\}^d$ s.t. $v_j[k] = 1$ iff $j \in V_k$

$(i, j) \in (U \times V) \cap C$ iff $(i, j) \in (U \times V)$ and $\neg \exists k: (i, j) \in U_k \times V_k$
iff $(i, j) \in (U \times V)$ and $< u_i, v_j > = 0$

Observation: To compute $(U \times V) \cap C$ we can list all $\leq K$ pairs $(i, j) \in (U \times V)$ such that u_i and v_j are orthogonal.

Reminder: OV Algorithm

Set $A := \{u_1, \dots, u_n\}$, $B := \{v_1, \dots, v_n\}$

1. Divide A and B into $q = \left\lceil \frac{n}{s} \right\rceil$ subsets of size $\leq s$:
   $A_1, \dots, A_q$ and $B_1, \dots, B_q$
2. Construct polynomial
   $P(a_1[1], \dots a_1[d], \dots, a_s[1] \dots, a_s[d], b_1[1], \dots b_1[d], \dots, b_s[1] \dots, b_s[1])$
   $P(A_i, B_j) = 1$ if and only if A_i, B_j contains orthogonal pair
   …with high probability
3. For every pair of subsets A_i, B_j: evaluate P on A_i, B_j
   …simultaneously! $\rightarrow O\left(\frac{n^2}{s^2} \text{ polylog}(n)\right)$
4. Return "yes" if some A_i, B_j contains orthogonal pair, "no" otherwise

To bound #monomials, set $s = 2^{\epsilon \log n}$ for sufficiently small ϵ (*)

New:

For every pair A_i, B_j containing orthogonal pair:
Report all orthogonal pairs by checking for every corresponding pair (i, j) if $(i, j) \in C$ (constant time lookup in matrix D!)

$O(Ks^2)$

(*) Requires tighter analysis than provided in lecture 3 (where we had $s = 2^{\epsilon \log n / \log d}$)

Variant of Orthogonal Vectors

OV Listing with Oracle: Given two sets of vectors $U \subseteq \{0,1\}^d, V \subseteq \{0,1\}^d$ containing at most K orthogonal pairs and an oracle supporting $O(1)$ time access to $< u, v >$ for any pair $u \in U$ and $v \in V$, report all orthogonal pairs in $U \times V$.

Time: $O\left(\frac{n^2}{s^2} \text{ polylog}(n) + Ks^2\right)$

Note: Without oracle we just get $O\left(\frac{n^2}{s^2}\, polylog(n) + Ks^2 d\right)$. This is not good enough in our application where we set $K = n^2/d$.

Algorithm Overview

1. Check for small submatrix
2. Check for dense submatrix
3. Check among previously seen pairs
4. Estimate number of unseen pairs
5. (a) If estimate is high, enumerate pairs and mark as seen
6. (b) If estimate is low, list unseen pairs

Parameters:

- $y \coloneqq n^{3/2}$
- $z \coloneqq 2^{\delta\sqrt{\log n}}$

Small Submatrix

1. Check for small submatrix

   If $|U| \times |V| < \frac{n^2}{z}$:

   Try all $i \in U, j \in V$

   If $M[i, j] = 1$ for some pair, then return 1

Dense Submatrix

2. Check for dense submatrix
   Sample y uniform random pairs $(i,j) \in U \times V$
   If $M[i,j] = 1$ for some pair, then return 1

Claim: If $M[U \times V]$ has $\geq \frac{cn^2 \log n}{y}$ 1-entries, then $M[i,j] = 1$ for some sample pair with probability at least $1 - \frac{1}{n^c}$.

Proof:

Probability that a sampled pair (i,j) is a 1-entry of $M[U \times V]$:

$$\frac{\#1 \text{ entries in } M[U\times V]}{|U\times V|} \geq \frac{cn^2 \log n}{y\,|U\times V|} \geq \frac{cn^2 \log n}{y\, n^2} = \frac{c \log n}{y}$$

Probability that no sampled pair (i,j) is a 1-entry of $M[U \times V]$:

$$\left(1 - \frac{c \log n}{y}\right)^{y} = \left(\left(1 - \frac{c \log n}{y}\right)^{\frac{y}{c \log n}}\right)^{c \log n} \leq \left(\frac{1}{e}\right)^{c \log n} = \frac{1}{n^c}$$

$$\boxed{\text{Fact: } \lim \left(1 - \frac{1}{x}\right)^{x} = e}$$

Previously Seen Pairs

3. Check among pairs seen before
   For all triples (U_k, V_k, S_k) in L and all pairs $(i, j) \in S_k$:
   If $(i, j) \in U \times V$, then return 1

Size Estimation

4. Estimate number of unseen pairs

   Goal: estimate size of $W = (U \times V) \cap C$

   $R :=$ sample of $\frac{n^2}{z}$ uniform random pairs from C

   $b := \frac{|R \cap (U \times V)|}{|R|} \cdot |C|$

   Efficient sampling from C: keep C in tree data structure (or similar)

Claim: $E[b] = |W|$

Proof: Random variables: $X_i = 1$ if i-th sample of R in $U \times V$

$X_i = 0$ otherwise

$\Pr[X_i = 1] = \frac{|W|}{|C|}$

$$E[b] = \frac{|C|}{|R|} \cdot E[|R \cap (U \times V)|] = \frac{|C|}{|R|} \cdot E\left[\sum_{i=1}^{|R|} X_i\right] = \frac{|C|}{|R|} \cdot \sum_{i=1}^{|R|} E[X_i]$$

$$= \frac{|C|}{|R|} \cdot \sum_{i=1}^{|R|} \frac{|W|}{|C|} = |W|$$

Chernoff Bound

Theorem: Let $X_1, \dots, X_t$ be a sequence of t independent Bernoulli trials s.t. $\Pr[X_i = 1] = p$ and $\Pr[X_i = 0] = 1 - p$ and $\mu := E\left[\sum_{i=1}^{t} X_i\right]$.

1. For every $\delta > 0$:
$$\Pr\left[\sum_{i=1}^{t} X_i \geq (1+\delta)\mu\right] \leq \exp\left(-\frac{\delta^2}{2+\delta}\mu\right)$$
2. For every $\delta \in [0,1]$:
$$\Pr\left[\sum_{i=1}^{t} X_i \leq (1-\delta)\mu\right] \leq \exp\left(-\frac{\delta^2}{2}\mu\right)$$

Applying Chernoff Bound

Theorem: Let $X_1, \dots, X_t$ be a sequence of t independent Bernoulli trials s.t. $\Pr[X_i = 1] = p$ and $\Pr[X_i = 0] = 1 - p$ and $\mu \coloneqq E\left[\sum_{i=1}^{t} X_i\right]$.

2. For every $\delta \in [0,1]$:

$$\Pr\left[\sum_{i=1}^{t} X_i \le (1-\delta)\mu\right] \le \exp\left(-\frac{\delta^2}{2}\mu\right)$$

Claim: Let $W = (U \times V) \cap C$. If $|W| > \frac{4n^2}{z}$, then $b > \frac{2n^2}{z}$ whp.

$$\mu = E\left[\sum_{i=1}^{t} X_i\right] = \frac{|R| \cdot |W|}{|C|} = \frac{n^2}{z} \cdot \frac{|W|}{|C|} \ge \frac{4n^4}{z^2\,|C|} \ge \frac{4n^2}{z^2} \ge 4\log^2 n \quad \left(z \le \frac{n}{\log n}\right)$$

$$\Pr\left[b < \frac{2n^2}{z}\right] = \Pr\left[|R \cap (U \times V)| < \frac{2n^2\,|R|}{z\,|C|}\right] = \Pr\left[\sum_{i=1}^{t} X_i < \frac{2n^2\,|R|}{z\,|C|}\right]$$

$$\le \Pr\left[\sum_{i=1}^{t} X_i < \frac{1}{2} \cdot \frac{|R| \cdot |W|}{|C|}\right] = \Pr\left[\sum_{i=1}^{t} X_i < \left(1 - \frac{1}{2}\right) \cdot \mu\right]$$

$$\le \exp\left(-\frac{1}{8}\mu\right) \le \exp\left(-\frac{\log^2 n}{2}\right) \le \exp(-\log n) = \frac{1}{n}$$

Applying Chernoff Bound

Theorem: Let $X_1, \ldots, X_t$ be a sequence of t independent Bernoulli trials with $\Pr[X_i = 1] = p$ and $\Pr[X_i = 0] = 1 - p$.

1. For every $\delta > 0$ and $\mu' \geq \mu$:

$$\Pr\left[\sum_{i=1}^{t} X_i \geq (1+\delta)\mu'\right] \leq \exp\left(-\frac{\delta^2}{2+\delta}\mu'\right)$$

Claim: Let $W = (U \times V) \cap C$. If $|W| \leq \frac{n^2}{z}$, then $b < \frac{2n^2}{z}$ whp

$$\mu' := \frac{n^2}{z} \cdot \frac{|R|}{|C|} \geq \frac{|R|}{z} = \frac{n^2}{z^2} \geq \log^2 n \qquad \left(z \leq \frac{n}{\log n}\right)$$

$$\mu = E\left[\sum_{i=1}^{t} X_i\right] = \frac{|R| \cdot |W|}{|C|} \leq \frac{n^2}{z} \cdot \frac{|R|}{|C|} = \mu'$$

$$\Pr\left[b \geq \frac{2n^2}{z}\right] = \Pr\left[\sum_{i=1}^{t} X_i \geq \frac{2n^2\,|R|}{z\,|C|}\right] = \Pr\left[\sum_{i=1}^{t} X_i \geq 2\mu'\right]$$

$$\leq \exp\left(-\frac{1}{3}\mu'\right) \leq \frac{1}{n}$$

Exhaustive Search

5. (a) If estimate is high, enumerate pairs and mark as seen

If $b > \frac{2n^2}{z}$:

- Compute answer to query (U, V)
- Determine $S = \{(i,j) \in U \times V \mid M[i,j] = 1\}$
  $|S| \leq \frac{cn^2 \log n}{y}$ (with high probability)
- Determine $W = (U \times V) \cap C$
  If $|W| < \frac{n^2}{z}$ or $|S| > \frac{cn^2 \log n}{Y}$: immediately return answer to query
  (happens with low probability)
- Add triple (U, V, S) to L
- Remove all $(i,j) \in U \times V$ from C
  (Zero out entries of D)
- Return answer to query (U, V)

List Unseen Pairs

6. (b) If estimate is low, list unseen pairs

If $b \leq \frac{2n^2}{z}$:

- Determine $W = (U \times V) \cap C$

  $|W| \leq \frac{4n^2}{z}$ (with high probability)
- Use OV Listing algorithm
- Check if there is $(i, j) \in W$ s.t. $M[i, j] = 1$

Running Time Analysis

Crucial observations:

1. Every time a triple (U, V, S) is added to the list, C is reduced by at least $\frac{n^2}{z}$
   $\Rightarrow$ length of list: $|L| \leq \frac{n^2}{n^2/z} = z$

2. Every time a triple (U, V, S) is added to the list, we have $|S| \leq O\left(\frac{n^2 \log n}{y}\right)$

First Part of Algorithm

1. Check for small submatrix $O\left(\frac{n^2}{z}\right)$

   If $|U| \times |V| < \frac{n^2}{z}$:

   Try all $i \in U, j \in V$

   If $M[i,j] = 1$ for some pair, then return 1

2. Check for dense submatrix $O(y)$

   Sample y uniform random pairs $(i,j) \in U \times V$

   If $M[i,j] = 1$ for some pair, then return 1

   Otherwise: Submatrix $M[U \times V]$ has $\leq \frac{cn^2 \log n}{y}$ 1-entries

3. Check among pairs seen before

   For all triples (U_k, V_k, S_k) in L and all pairs $(i,j) \in S_k$:

   If $(i,j) \in U \times V$, then return 1

$$O\left(\sum_{k=1}^{|L|} |S_k|\right) \leq O\left(\sum_{k=1}^{|L|} \frac{n^2 \log n}{y}\right) \leq O\left(\frac{z\, n^2 \log n}{y}\right)$$

4. Estimate number of unseen pairs $O\left(\frac{n^2}{z}\right)$

   $R :=$ sample of $\frac{n^2}{z}$ uniform random pairs from C

   $b := \frac{|R \cap (U \times V)|}{|R|} \cdot |C|$

Exhaustive Search

5. (a) If estimate is high, enumerate pairs and mark as seen

If $b > \frac{2n^2}{z}$:

- Compute answer to query (U, V)
- Determine $S = \{(i,j) \in U \times V \mid M[i,j] = 1\}$
  $|S| \leq \frac{cn^2 \log n}{y}$ (with high probability)
- Determine $W = (U \times V) \cap C$
  If $|W| < \frac{n^2}{z}$ or $|S| > \frac{cn^2 \log n}{Y}$: immediately return answer to query
  (happens with low probability)
- Add triple (U, V, S) to L
- Remove all $(i,j) \in U \times V$ from C
  (Zero out entries of D)
- Return answer to query (U, V)

$O(n^2)$

Expensive, but can be amortized!
Executed at most z times

List Unseen Pairs

6. (b) If estimate is low, list unseen pairs

If $b \leq \frac{2n^2}{z}$:

- Determine $W = (U \times V) \cap C$

  $|W| \leq \frac{4n^2}{z}$ (with high probability)
- Use OV Listing algorithm with table-lookup oracle
- Check if there is $(i, j) \in W$ s.t. $M[i, j] = 1$

$$O\left(\left(\frac{n}{s}\right)^2 + \frac{n^2}{z} \cdot s^2\right)$$

Total Running Time

q: number of queries

Check for dense submatrix

$$O\left(q\cdot\left(\frac{n^2}{z}+y+\frac{z\,n^2}{y}+\overbrace{\left(\frac{n}{s}\right)^2+\frac{n^2}{z}\cdot s^2}^{\text{List unseen pairs}}\right)\mathrm{polylog}(n)+zn^2\right)$$

Check for small submatrix / Size estimation

Check among previously seen pairs

Exhaustive search

Parameter choice:

- $y = n^{3/2}$
- $z = 2^{\delta\sqrt{\log n}}$ (for some $\delta > 0$)
- $s = 2^{\epsilon\delta\sqrt{\log n}}$ (for sufficiently small $\epsilon > 0$)

$\Rightarrow$ amortized $O\left(n^2/2^{\delta\sqrt{\log n}}\right)$ per query

max planck institut
informatik

Summary

1. Algorithmic use of OMv to OuMv reduction
2. Essentially: OuMv to OV reduction
3. 3 sources of randomization:
   - Hitting set
   - Size estimation
   - Probabilistic polynomial