Fine-grained complexity of orthogonal vectors problem and it's variants

OV, OV', Max IP+, HS, ...

SUSHMA KAMUJU

Department of Computer Science and Engineering,
IIT Palakkad, Kerala

May 15, 2024

Motivation

1. Linear time - best runtime
2. Polynomial time - 2nd best.
3. In polynomial time problems, quadratic, cubic, ...
4. Cannot be solved in linear time or
5. Design better algorithms.

   (or)

   Show that solving a problem in better than best known can improve the runtime of many others.
6. Polynomial time reduction.

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Polynomial time reduction.

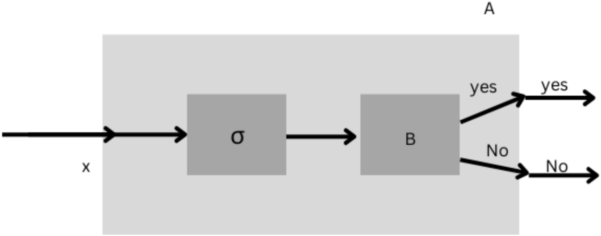


Figure: $A \leq B$

1. $A \leq_m^p B$
2. If B can be solved in polynomial time A can also be.

Fine-grained reduction

1. Polynomial time reduction σ, Fine-grained Reduction σ.
2. If B can be solved sub quadratic time then σ has be done in sub quadratic for A to be solved in sub quadratic time.

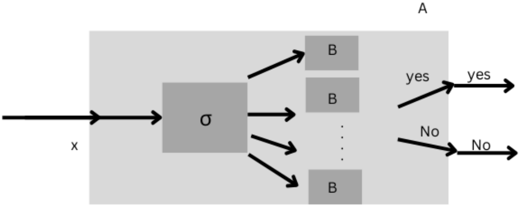


Figure: $A \leq B$

Orthogonal Vectors Problem (OV)

1. **Input:** Collections A, B, of n vectors each from $\{0, 1\}^d$.
   Output: yes, if there exists distinct vectors $v_1 \in A$, $v_2 \in B$ such that $\langle v_1, v_2 \rangle = 0$, and no otherwise.
2. **Example:**$n = 3$, $d = 4$
   $A = \{ [1,0,1,0], [1, 0, 1, 0], [1, 1, 1, 0]\}$
   $B = \{[0, 0, 1, 1], [1, 1, 1, 1], [0,1,0,1] \}$
   For this example, OV returns yes.
3. **Example:** $n = 3$, $d = 4$
   $A = \{[1, 1, 1, 0], [1, 1, 1, 0], [1, 1, 1, 0]\}$
   $B = \{[0, 1, 0, 1], [1, 1, 1, 1], [1, 1, 1, 1]\}$
   For this example, OV returns no.

OV

1. Brute-force algorithm - $O(n^2 \cdot d)$ time
2. Another algorithm - $O(n \cdot 2^d \cdot d)$ time.
3. Run time of Orthogonal Vectors Problem- $O(n^2 \log n)$ for any d.

Orthogonal Vector Hypothesis

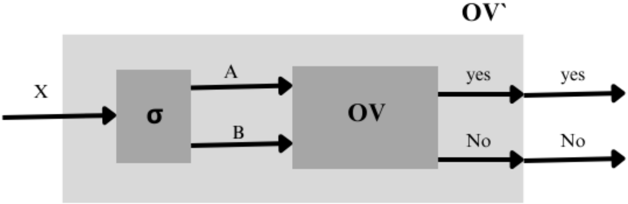
For any $\epsilon > 0$ there cannot be an algorithm running in $O(n^{2-\epsilon} \cdot poly(d))$ time solving OV.

4. Last Semester: 3 SUM has this hypothesis that it cannot be solved in $O(n^{2-\epsilon})$ where $\epsilon > 0$.

1. $OV' \leq_f OV$
2. $OV \leq_f OV'$
3. $OV \leq_f OVSearch$
4. $OVSearch \leq_f OV$
5. $\mathrm{HS} \leq_f \mathrm{OV}Search$

$OV' \leq OV$

1. **Input:** A collection X of n vectors from $\{0,1\}^d$.
   Output: yes, if there exits distinct vectors $v_1 \in X$, $v_2 \in X$ such that $\langle v_1, v_2 \rangle = 0$, and no otherwise.



$OV \leq OV'$

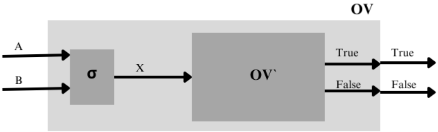


Figure: $OV \leq \ OV'$

1. construct $A' \subseteq \{0,1\}^{d+2}$ where $A' = \{(a_1, \ldots, a_d, 1, 0) \mid (a_1, \ldots, a_d) \in A\}$
2. similarly, $B' = \{(b_1, \ldots, b_d, 0, 1) \mid (b_1, \ldots, b_d) \in B\}$.
3. The output of σ, X= $A' \cup B'$
4. X is passed to OV' which returns true if there are vectors in it with dot product 0 and no otherwise.

$OV \leq OVSearch$

1. **Input:** Collections A, B, of n vectors each from $\{0,1\}^d$.
   Output: Output $(v_1, v_2) \in A \times B$ such that $\langle v_1, v_2 \rangle = 0$, if it exists and null set otherwise.

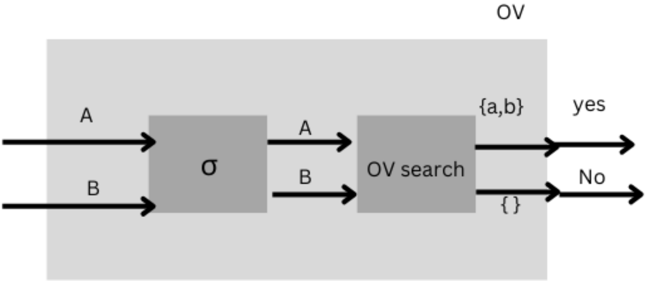


Figure: OV $\leq$ OV Search

$OVSearch \leq OV$

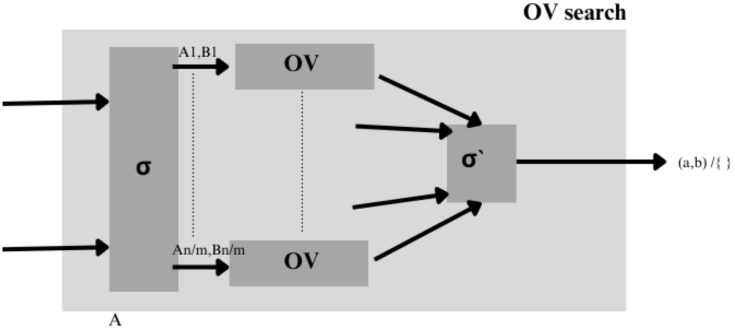
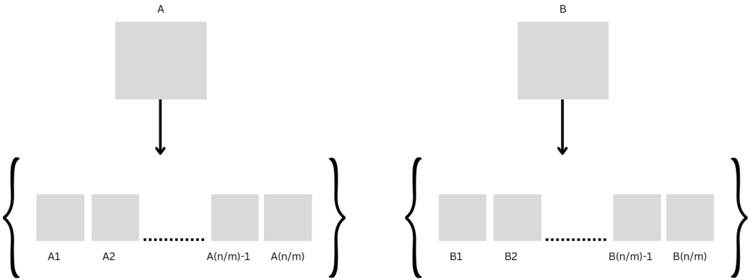


Figure: OV Search$\leq$ OV

Reduction

1. divide A into smaller sets each of size m (where $m < n$, which will be fixed later) arbitrarily.
2. Repeat the same for B



Reduction

1. does brute force searches over all pairs A_i and B_j to find it any one of them have an orthogonal pair using the an algorithm for OV.
2. This involves calling the OV algorithm $\left(\frac{n}{m}\right)^2$.
3. If one of the pairs A_i, B_j returns a yes, we brute-force over all elements in A_i, B_j and find the orthogonal pair.
4. If no pair was found, we return that there is no pair. This takes $O(m^2d)$ time.

Time Complexity

1. We start by assuming that there is a deterministic algorithm that solved OV in time $OV(n, d) := n^{2-\epsilon}(d)$ time for some $\epsilon > 0$.
2. We then show that OV*Search* can be solved in $n^{2-2\epsilon}(d)$ time.
3. $(\frac{n}{m})^2 \cdot \mathrm{OV}(m, d) + m^2 \cdot d(brute - force) + O(nd)$.
4. $$\begin{aligned}\left(\frac{n}{m}\right)^2 \cdot m^{2-\epsilon} \cdot \mathrm{poly}(d) + m^2 \cdot d \leq &\left(\frac{n}{m}\right)^2 \cdot m^{2-\epsilon} \cdot \mathrm{poly}(d) + m^2 \cdot \mathrm{poly}(d) \\ \leq &(\left(\frac{n}{m}\right)^2 \cdot m^{2-\epsilon} + m^2)\mathrm{poly}(d)\end{aligned}$$
5. $m = n^{\left(\frac{2}{2+\epsilon}\right)}$, the time complexity is $n^{(2-2\epsilon)} \cdot \mathrm{poly}(d)$

Hitting Set HS

1. **Input:** Collections A, B, of n vectors each from $\{0,1\}^d$.
   Output: yes, if there exists $v_1 \in A$ for all $v_2 \in B$ such that $\langle v_1, v_2 \rangle \neq 0$, and no otherwise.

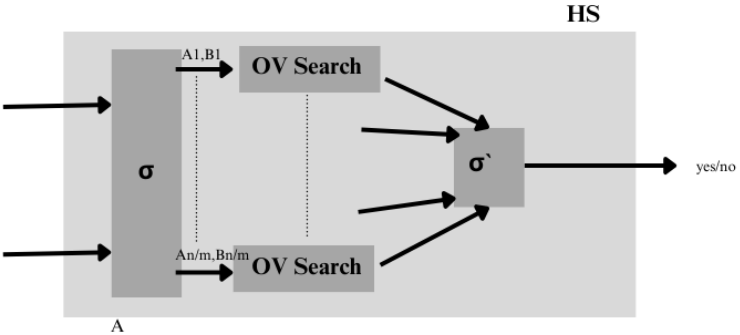
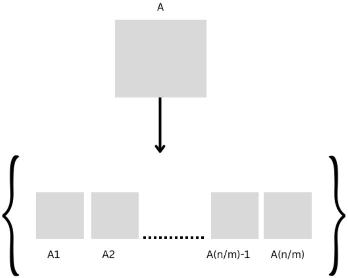


Figure: HS $\leq$ OV*Search*

Reduction

1. divide A into smaller sets each of size m (where $m < n$, which will be fixed later) arbitrarily.
2. Repeat the same for B



1. each $i \in \{1, 2, \ldots, n/m\}$ and for each $j \in \{1, 2, \ldots, n/m\}$, do OV*Search* on the pair (A_i, B_j)
2. removes all the vectors in A_i (to get A_i') for which the inner product with some element in B_j is zero
3. This can be found from the result of *OVSearch*
4. We repeat this for all the B_j's
5. Once all the B_j's are checked, if the resulting A_i is non-empty, return yes.Else proceed to the next A_i
6. At the end, if all A_i's turned out to be empty, return no.

$$\begin{array}{cc} & \begin{array}{cccc} B1 & B2 & \cdots & B(n/m) \end{array} \\ A1 & \begin{bmatrix} \text{A1 - OV Search(A1,B1)} & \text{A1 - OV Search(A1,B2)} & \cdots & \text{A1 - OV Search(A1,B3)} \end{bmatrix} \end{array} = A1$$

$$\begin{array}{cc} & \begin{array}{cccc} B1 & B2 & \cdots & B(n/m) \end{array} \\ A2 & \begin{bmatrix} \text{A2 - OV Search(A2,B1)} & \text{A2 - OV Search(A2,B2)} & \cdots & \text{A2 - OV Search(A2,B3)} \end{bmatrix} \end{array} = A2$$

$$\vdots$$

$$\begin{array}{cc} & \begin{array}{cccc} B1 & B2 & \cdots & B(n/m) \end{array} \\ An/m & \begin{bmatrix} \text{An/m - OV Search(A2,B1)} & \text{An/m - OV Search(A2,B2)} & \cdots & \text{An/m - OV Search(A2,B3)} \end{bmatrix} \end{array} = An/m$$

Time complexity

1. $m \cdot \text{OV Search}(m, d) \cdot \left(\frac{n}{m}\right)$ instead OV Search$(m, d) \cdot m \cdot \left(\frac{n}{m}\right)^2$
2. $$\begin{aligned} n \cdot m^{2-\epsilon} \cdot \text{poly}(d) + n &\leq n \cdot m^{2-\epsilon} \cdot \text{poly}(d) + n \cdot \text{poly}(d) \\ &= (n \cdot m^{2-\epsilon} + n) \cdot \text{poly}(d) \end{aligned}$$
3. Choosing $m = n^{\frac{1}{1+\left(\frac{1}{1-\epsilon}\right)}}$, we get the above runtime to be $O(n^{2-\epsilon}(d))$.

1. Non-Orthogonal Vectors NOV
   Input: Collections A, B, of n vectors each from $\{0, 1\}^d$.
   Output: yes, if there exists distinct vectors $v_1 \in A$, $v_2 \in B$ such that $\langle v_1, v_2 \rangle \neq 0$, and no otherwise.
2. Always Orthogonal AOV
   Input: Collections A, B, of n vectors each from $\{0, 1\}^d$.
   Output: yes, if there exists $v_1 \in A$ for all $v_2 \in B$ such that $\langle v_1, v_2 \rangle = 0$, and no otherwise.

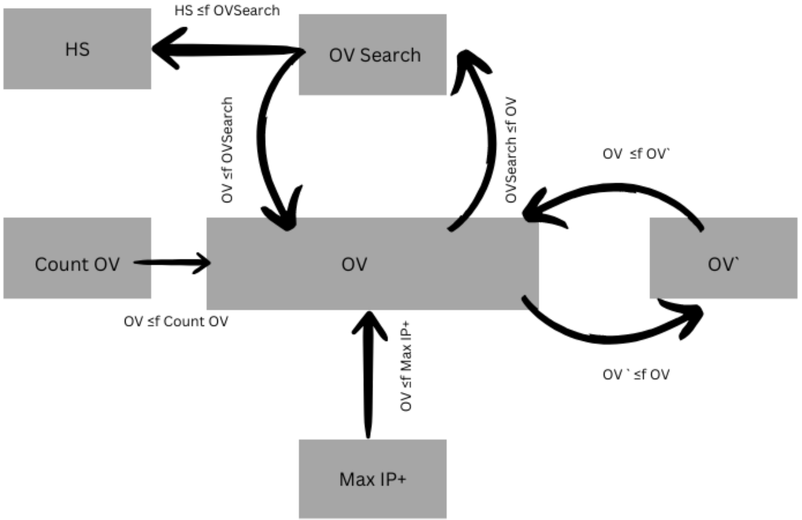
variants of OV

Problem	Quantification on $v_1 \in A$	Quantification on $v_2 \in B$	$\langle v_1, v_2 \rangle$ check
OV	$\exists$	$\exists$	$\langle v_1, v_2 \rangle = 0$
NOV	$\exists$	$\exists$	$\langle v_1, v_2 \rangle \neq 0$
HS	$\exists$	$\forall$	$\langle v_1, v_2 \rangle \neq 0$
AOV	$\exists$	$\forall$	$\langle v_1, v_2 \rangle = 0$

Table: Variants of Orthogonal Vectors Problem

1. In total, there should have been eight variants, but the other four variants will essentially be complement of the above four problems and hence are not listed.

Results


HS
HS ≤f OVSearch
OV Search
OV ≤f OVSearch
OVSearch ≤f OV
OV ≤f OV`
Count OV
OV
OV`
OV ≤f Count OV
OV ≤f Max IP+
OV ` ≤f OV
Max IP+

1. MaxIP+ $\leq$ OV
2. NOV
3. AOV

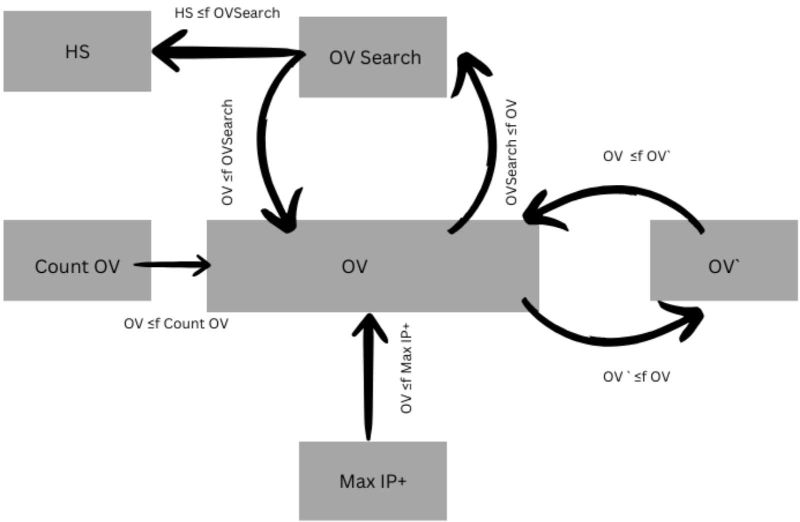
THANK YOU

SUSHMA KAMUJU
112001014

Orthogonal vectors Problem and its varinats

1. **Orthogonal Vectors OV**
2. **Non-Orthogonal Vectors NOV**
3. **Hitting Set HS**
4. **Always Orthogonal AOV**
5. **OV′**
6. **Maximum Inner Product MaxIP**
7. **Minimum Inner Product MinIP**
8. **Maximum Inner Product MaxIP+**
9. **Search Orthogonal Vectors OVSearch**
10. **Count Orthogonal Vectors countOV**

Fine-Grained Reductions



THANK YOU

SUSHMA KAMUJU
112001014

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Let n, d be non-negative integers.

- **Input:** A collection X of n vectors from $\{0,1\}^d$.
- **Output:** yes, if there exits distinct vectors $v_1 \in X$, $v_2 \in X$ such that $\langle v_1, v_2 \rangle = 0$, and no otherwise.

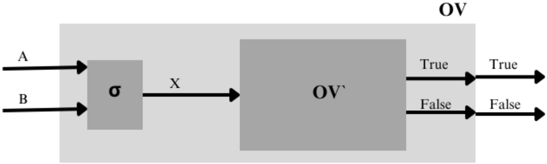


Figure: $OV \leq \; OV'$

Working of reduction

1. construct $A' \subseteq \{0,1\}^{d+2}$ where
   $A' = \{(a_1, \dots, a_d, 1, 0) \mid (a_1, \dots, a_d) \in A\}$
2. similarly, $B' = \{(b_1, \dots, b_d, 0, 1) \mid (b_1, \dots, b_d) \in B\}$.
3. The output of σ, $X = A' \cup B'$
4. X is passed to OV' which returns true if there are vectors in it with dot product 0 and no otherwise.

Proof Of Reduction

(A, B) is an yes instance of $OV \iff X$ (where $X = \sigma(A, B)$) is an yes instance of OV'

Forward direction

- suppose (A, B) is a yes instance of OV. Hence, there exists $v_1 \in A$ and $v_2 \in B$ such that $\langle v_1, v_2 \rangle = 0$.
- By the map σ defined above, there exists v_1' and v_2' in X where v_1' is v_1 concatenated with $(1, 0)$ and v_2' is v_2 concatenated with $(0, 1)$.
- By construction, $\langle v_1', v_2' \rangle = \langle v_1, v_2 \rangle$ Since $\langle v_1, v_2 \rangle = 0$, we can conclude that X is a yes instance of OV'.

Reverse direction

1. Suppose X is a 'yes' instance of OV', indicating there exist vectors v_1' and v_2' such that their dot product is 0.
2. Let v_1' be (v_1, a, b) and v_2' be (v_2, c, d), where a, b and $c, d \in \{(0,1),(1,0)\}$.
3. Since $\langle v_1', v_2' \rangle = 0$, then $\langle (v_1, (a,b)), (v_2, c, d) \rangle = 0$.
4. This expands to $\langle v_1, v_2 \rangle + \langle (a,b), (c,d) \rangle = 0$.
5. To ensure the whole term is 0, as both terms are positive, the only possibility is to make both terms 0.
6. Therefore, $\langle v_1, v_2 \rangle = 0$ and $\langle (a,b), (c,d) \rangle = 0$.
7. For this to be 0, (a,b) and (c,d) should be different.
8. This implies if $v_1 \in A$, then $v_2 \in B$, and if $v_2 \in A$, then $v_1 \in B$.

$$OV' \leq_f OV$$

1. How reduction works

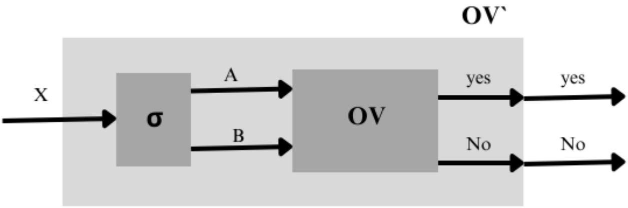


Figure: $OV \leq OV'$

How Reduction works

X is an yes instance of $OV' \iff (A, B)$ where $(A, B) = \sigma(X)$ is an yes instance of OV

Forword direction

1. Consider an yes instance X of OV'. This implies there is pair of vectors $v_1, v_2 \in X$ whose inner product is 0.
2. Then there will be pair of vectors in $A \times B$ such that their inner product is 0.
3. Hence A, B is an yes instance of OV.

Reverse direction

1. Suppose X is no instance of OV', then there are no pair of vectors in X that have inner product as 0,
2. in which case it is not possible to get any pair of vectors in $A \times B$ whose inner product is 0.
3. Hence A, B is no instance of OV.

OV Search $\leq$ OV

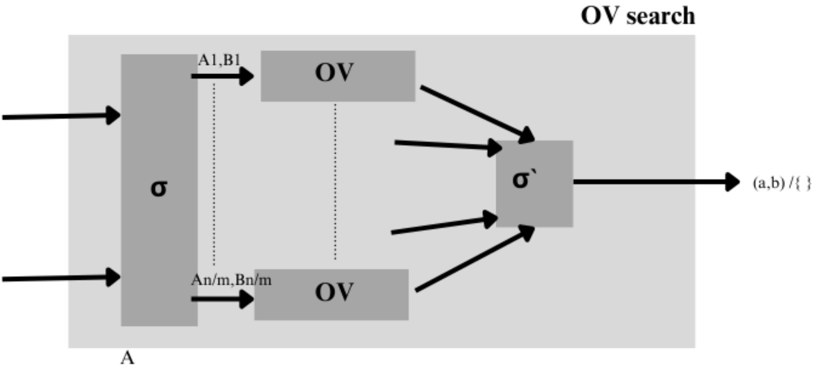
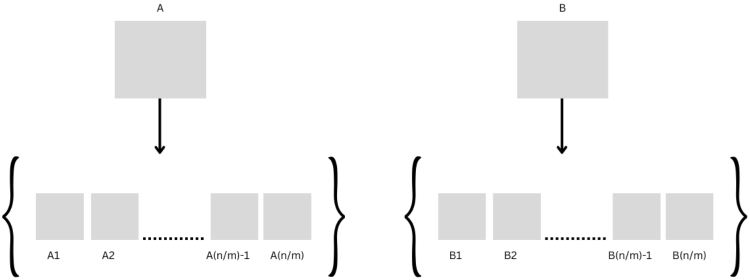


Figure: OV Search$\leq$ OV

How reduction works

Step1:

1. divide A into smaller sets each of size m (where $m < n$, which will be fixed later) arbitrarily.
2. Repeat the same for B



How reduction works

Step2:

- does brute force searches over all pairs A_i and B_j to find it any one of them have an orthogonal pair using the an algorithm for OV.
- This involves calling the OV algorithm $\left(\frac{n}{m}\right)^2$.
- If one of the pairs A_i, B_j returns a yes, we brute-force over all elements in A_i, B_j and find the orthogonal pair.
- If no pair was found, we return that there is no pair. This takes $O(m^2 d)$ time.

Proof of reduction

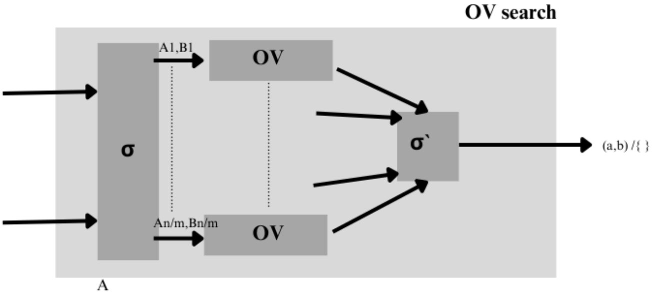


Figure: OV Search $\leq$ OV

$$(A, B) \in \ OVSearch \iff \exists i, j(A_i, B_j) \text{ is a yes instance of } \ OV.$$

Forward direction proof:

1. Consider yes instance of OV. This implies there is pair of vectors, say, v_1, v_2 where $v_1 \in A$ and $v_2 \in B$ such that their dot product is 0.
2. After dividing A, B into smaller collections let $v_1 \in A_i$, $V_2 \in B_j$. Hence $(A_i, B_j) \in OV$, yes instance.

Reverse direction proof:

1. Consider a no instance (A, B) of OV. This implies these exists no pair of vectors in (A, B) such that their dot product is 0.
2. Hence none of the pairs in $\{A_1, A_2, \ldots, A_{\frac{n}{m}}\} \times \{B_1, B_2, \ldots, B_{\frac{n}{m}}\}$ belongs to OV, no instance.

OV $\leq$ MaxIP+

Let n, d be non-negative integers.

1. **Input:** Collections A, B, of n vectors each from $\mathbb{R}^d_{\geq 0}$.
2. **Output:** Compute $\max_{(v_1, v_2) \in A \times B} \langle v_1, v_2 \rangle$.

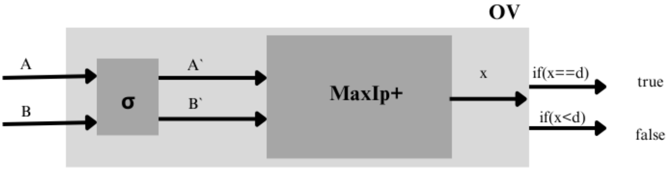


Figure: MaxIp+ $\leq$ OV

How reduction works

1. construct $A' \in \{0,1\}^{3d}$ where A' is given as for all $a \in A$, $a' \in A'$ so a' is given by $[a, 1-a, 1-a]$ so the length of each vector in A' becomes 3d where the length of each vector in A is d.
2. for all $b \in B$, $b' \in B'$ such that b' is given by $[1-b, 1-b, b]$ so the length of each vector in B' becomes 3d where the length of each vector in B is d.
3. $MaxIP+$ returns the maximum inner product for the sets A' and B'

Proof of Reduction

(A, B) is a yes instance of $OV \iff$ MAXIP+ instance of (A', B') (where $(A', B') = \sigma(A, B)$) returns d.

Forward Direction

1. Consider A, B as yes instances of OV. This implies there is a pair of vectors, let $a \in A$ and $b \in B$, whose dot product is 0 $\langle a, b \rangle = 0$.
2. Let $a' \in A'$ and $b' \in B'$ where $a' = a||\overline{a}||\overline{a}$ and $b' = \overline{b}||\overline{b}||b$.
3. $\langle a', b' \rangle = d - \langle a, b \rangle = d$ (lemma proved in report)

Reverse Direction

1. Consider no instance of OV this means there is no pair of vectors from A, B whose dot product is 0,this implies $\forall a \in A, \forall b \in B, \langle a, b \rangle > 0$.
2. from lemma$\langle a', b' \rangle = d - \langle a, b \rangle$ as $\langle a, b \rangle > 0$ there is no possibility that $\langle a', b' \rangle$ can ever become d.

$$HS \leq_f OVSearch$$

Hitting Set HS

Let n, d be non-negative integers.

- **Input:** Collections A, B, of n vectors each from $\{0, 1\}^d$.
- **Output:** yes, if there exists $v_1 \in A$ for all $v_2 \in B$ such that $\langle v_1, v_2 \rangle \neq 0$, and no otherwise.

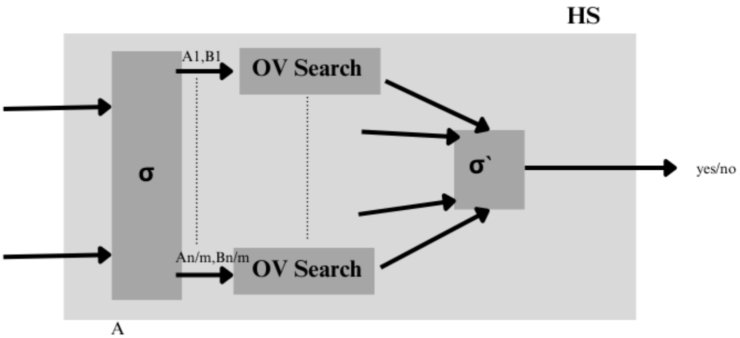
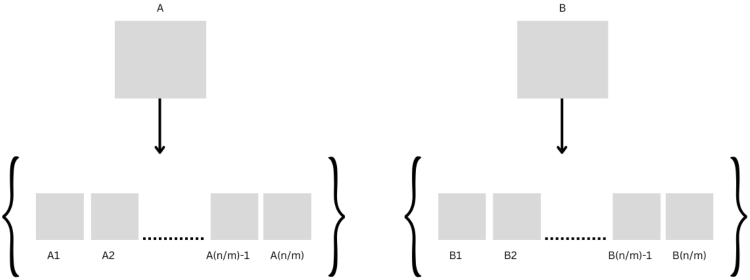


Figure: $HS \leq\ OVSearch$

How reduction works

Step1:

1. divide A into smaller sets each of size m (where $m < n$, which will be fixed later) arbitrarily.
2. Repeat the same for B



How reduction works

Step2:

- each $i \in \{1, 2, \ldots, n/m\}$ and for each $j \in \{1, 2, \ldots, n/m\}$, do OV*Search* on the pair (A_i, B_j)
- removes all the vectors in A_i (to get A_i') for which the inner product with some element in B_j is zero
- This can be found from the result of *OVSearch*
- We repeat this for all the B_j's
- Once all the B_j's are checked, if the resulting A_i is non-empty, return yes.Else proceed to the next A_i
- At the end, if all A_i's turned out to be empty, return no.

$$
\begin{array}{cc}
 & \begin{array}{cccc} B1 & B2 & \cdots & B(n/m) \end{array} \\
A1 & \begin{bmatrix} \text{A1 - OV Search(A1,B1)} & \text{A1 - OV Search(A1,B2)} & \cdots & \text{A1 - OV Search(A1,B3)} \end{bmatrix}
\end{array}
= A1
$$

$$
\begin{array}{cc}
 & \begin{array}{cccc} B1 & B2 & \cdots & B(n/m) \end{array} \\
A2 & \begin{bmatrix} \text{A2 - OV Search(A2,B1)} & \text{A2 - OV Search(A2,B2)} & \cdots & \text{A2 - OV Search(A2,B3)} \end{bmatrix}
\end{array}
= A2
$$

$$\vdots$$

$$
\begin{array}{cc}
 & \begin{array}{cccc} B1 & B2 & \cdots & B(n/m) \end{array} \\
An/m & \begin{bmatrix} \text{An/m - OV Search(A2,B1)} & \text{An/m - OV Search(A2,B2)} & \cdots & \text{An/m - OV Search(A2,B3)} \end{bmatrix}
\end{array}
= An/m
$$

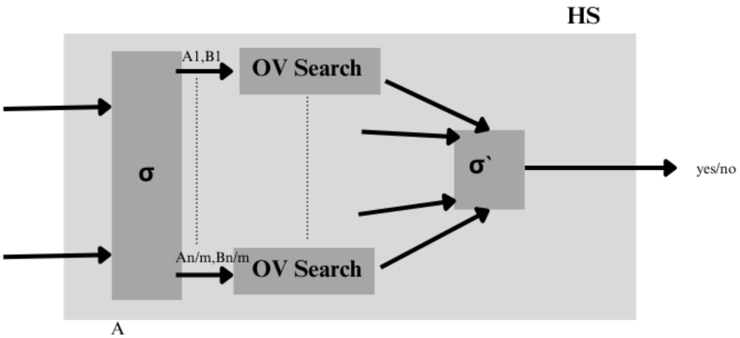


Figure: $HS \leq \ OVSearch$

Proof of reduction

A, B is yes instance of $HS \iff \exists A_i$ such that $\forall j$ the OV search on (A_i, B_j) returns a null set.

Forward direction

- Consider a yes instance of HS, this implies let $v_1 \in A$ such that for all $v_2 \in B$, $\langle v_1, v_2 \rangle \neq 0$.
- Hence, $v_1 \in A_i$ for some i. Hence, for any $j \in \{1, 2, \ldots, n/m\}$, if we perform an OV*Search* on (A_i, B_j), it is bound to return a null set by the hitting set property.

Reverse direction

- Let (A, B) be a no instance of HS. Then, for any $v_1 \in A$, there exists $v_2 \in B$ such that $\langle v_1, v_2 \rangle = 0$.
- Pick any A_i and any $v_1 \in A_i$.
- By the above statement, there exists a B_j such that for some $v_2 \in B_j$ the inner product of v_1 and v_2 is 0.
- Hence OV*Search* on the pair (A_i, B_j) will return (v_1, v_2) and hence is not a null set.

Time complexity and analysis

Conclusion

1. OV $\leq$ Count OV
2. $OV \leq_f OVsearch$
3. $MaxIP+ \leq OV$
4. The problems NOV and AOV