A Double-Exponential Lower Bound for the Distinct Vectors Problem*

Marcin Pilipczuk  Manuel Sorge

University of Warsaw, Poland

received 5th Feb. 2020, revised 6th July 2020, accepted 2nd Sep. 2020.

In the (binary) DISTINCT VECTORS problem we are given a binary matrix $A$ with pairwise different rows and want to select at most $k$ columns such that, restricting the matrix to these columns, all rows are still pairwise different. A result by Froese et al. [JCSS] implies a $2^{O(k)} \cdot \text{poly}(|A|)$-time brute-force algorithm for DISTINCT VECTORS. We show that this running time bound is essentially optimal by showing that there is a constant $c$ such that the existence of an algorithm solving DISTINCT VECTORS with running time $2^{O(2^c k)} \cdot \text{poly}(|A|)$ would contradict the Exponential Time Hypothesis.

**Keywords:** feature selection, data mining, computational complexity, parameterized algorithms

1 Introduction

For each $n \in \mathbb{N}$, let $[n] = \{1, \ldots, n\}$. Let $\Sigma$ be a set and $n, m \in \mathbb{N}$. By $\Sigma^{m \times n}$ we denote the set of $m$-row $n$-column matrices with entries in $\Sigma$. Let $A \in \Sigma^{m \times n}$. By $A[i, j]$ we denote the entry of $A$ in the $i$-th row and $j$-th column. By $A[i, *]$ and $A[*, j]$ we denote the $i$-th row and the $j$-th column of $A$, respectively. For easier notation, we often identify rows or columns and their indices. Let $I \subseteq [m]$ and $J \subseteq [n]$. By (i) $A[I, J]$, (ii) $A[I, *]$, and (iii) $A[*, J]$ we denote the submatrix of $A$ containing (i) only the entries that are simultaneously in rows in $I$ and columns in $J$, (ii) only the entries in rows in $I$, and (iii) only the entries in columns in $J$, respectively.

We study the computational complexity of the following decision problem.

**DISTINCT VECTORS**

**Instance:** A binary matrix $A \in \{0, 1\}^{m \times n}$ and $k \in \mathbb{N}$.

**Question:** Is there a subset $K \subseteq [n]$ of at most $k$ columns such that the rows in $A[*, K]$ are pairwise distinct?

We also say that $K$ as above is a solution.

DISTINCT VECTORS is a fundamental problem which has arisen in several different contexts. Notably, it has applications in database theory, where it models key selection in relational databases (e.g. [BFS17]), machine learning, where it models combinatorial feature selection [Cha+00], and in rough set theory, where it models finding some minimal structure [Paw91]. See Froese [Fro18] for an overview over the literature. We note that DISTINCT VECTORS is sometimes formulated with larger alphabet size than two, that is, the entries of $A$ may be more than two distinct symbols. Since we focus here on a lower bound, however, the binary formulation is sufficient for us. Froese et al. [Fro+16, Theorem 12] gave a problem kernel with size $2^{O(k)}$ for DISTINCT VECTORS parameterized by $k$. (A problem kernel with respect to a parameter $k$ is a polynomial-time self-reduction with an upper bound, a function of $k$, on the resulting instance size.) Simple brute force on the resulting instances yields a $2^{O(k)} \cdot \text{poly}(|A|)$-time algorithm for DISTINCT VECTORS. It is natural to ask whether this running time bound can be improved. Here, we answer this question negatively by proving the following.

**Theorem 1.** For each $\epsilon > 0$, if there is a $2^{O(2^c k)} \cdot \text{poly}(n + m)$-time algorithm solving DISTINCT VECTORS, then the Exponential Time Hypothesis is false, where $c = c(\epsilon) = 1/2 - \epsilon$.

*This research is part of a project that has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (Grant Agreement 714704).
Informally, the Exponential Time Hypothesis (ETH) states that 3SAT on $n$-variable formulas cannot be solved in $2^{o(n)}$ time [IP01]. Formally, we rely on the following formulation that comes from an application of the Sparsification Lemma [IPZ01].

**Conjecture 2** (Exponential Time Hypothesis + Sparsification Lemma). There exist constants $\delta, C > 0$ such that there is no algorithm that, given as input a 3CNF-SAT formula $\phi$ with $n$ variables and at most $C \cdot n$ clauses, runs in time $O(2^h)$ and correctly verifies the satisfiability of $\phi$.

The proof of Theorem 1 is given in Section 2. Herein, to simplify notation, we often write vectors $(v_1, \ldots, v_n) \in \Sigma^n$ as $v_1 v_2 \ldots v_n$. We also use $\circ \cdot$ to denote concatenation. That is, for each $n, m \in \mathbb{N}$ and each $(v_i)_{i \in [m]} \in \Sigma^n$ and $(w_i)_{i \in [m]} \in \Sigma^m$ we define $v_1 v_2 \ldots v_n \circ w_1 w_2 \ldots w_m = v_1 v_2 \ldots v_n w_1 w_2 \ldots w_m \in \Sigma^{n+m}$. Furthermore, for each $i \in \mathbb{N}$ and $\sigma \in \Sigma$ we define $\sigma(i) = \sigma \sigma \ldots \sigma \in \Sigma^i$. By log we refer to the base-two logarithm. By poly we refer to an arbitrary fixed polynomial.

## 2 Proof of Theorem 1

Let $\epsilon > 0$. Let $\delta$ and $C$ be the constants of Conjecture 2. Let $\phi$ be a boolean formula $\phi$ in conjunctive normal form with $r$ variables and $s$ clauses such that each clause has size exactly three and such that $s \leq C \cdot r$.

Below we construct an instance $(A, k)$ of DISTINCT VECTORS which has a solution if and only if $\phi$ is satisfiable and such that $A$ has $n = 2^{O(r/\log r)}$ columns and $m = O(r)$ rows, and there are $k \leq c' + 2 \log r$ columns to select for some constant $c'$. The construction can be carried out in $2^{O(r/\log r)}$ time. Thus, an algorithm solving DISTINCT VECTORS with running time $2^{O(2^{O(r/\log r)})} \cdot \text{poly}(n+m)$ for some constant $c$ can be used to check satisfiability of $\phi$ in time $2^{O(r/\log r)} + 2^{O(2^{O(r/\log r)})} \cdot \text{poly}(2^{O(r/\log r)} + O(r))$. Since $c = 1/2 - \epsilon$, this is $2^{o(r)}$ time, implying that the ETH is false.

**Construction.** Let $X = \{x_1, x_2, \ldots, x_r\}$ be the set of variables in $\phi$ and $C = \{C_1, C_2, \ldots, C_s\}$ the set of clauses. Without loss of generality, assume that $r$ is a power of two and $s$ equals $2^l - 1$ for some $l \in \mathbb{N}$. Otherwise, introduce variables that do not occur in any clause and repeat clauses as necessary. Note that this can be done in such a way that, afterwards, still $s = O(r)$. Let $r' := [r/\log r]$. We partition the variables into $\log r$ bundles $B_i = \{b_{i1}, b_{i2}, \ldots, b_{ir'}\} \subseteq X, i \in [\log r]$, where each bundle $B_i$ contains exactly $r'$ variables (repeat variables from the bundle if necessary to fill a bundle).

The columns of matrix $A$ are partitioned into $\log(r) + 1$ parts, one consistency part and one part for each bundle. The consistency part contains $\ell = \log(s + 1)$ columns. We will make sure that all of them can be assumed to be in the solution. In this way, these columns will serve to distinguish some rows corresponding to clause gadgets from each other. The remaining $\log r$ parts of columns correspond one-to-one to the bundles. The columns corresponding to $B_i$ are $B_i$'s columns. For each $i \in [\log r]$, there will be $\rho := 2^{r'}$ columns belonging to $B_i$ which correspond one-to-one to the possible truth-assignments to the variables in $B_i$. We will ensure that exactly one of the columns of $B_i$ will be chosen in any solution, that is, the solution chooses a truth-assignment to the variables in $B_i$.

We now describe the construction of $A$ by defining its rows. The rows of matrix $A$ are partitioned into two parts $I_1, I_2 \subseteq [m]$.

Recall $\rho = 2^{r'}$. The first part, $A[I_1, \ast]$, of the rows of $A$ consists of $\log r + 1$ rows, that is $I_1 = [\log r + 1]$. The first row, $A[1, \ast]$, contains only zeros. The $(i + 1)$-th row, $i \in [\log r]$, is defined by

$$A[i + 1, \ast] = 0^{(\log(s+1))} \circ 1^{((i-1)\rho)} \circ 0^{(\log r-i)\rho}.$$ 

That is, for each bundle $B_i$ there is a row which has 1 in the columns $\log(s+1) + (i-1)\rho + 1$ to $\log(s+1) + i\rho$ and 0 otherwise. We say that the columns $\log(s+1) + (i-1)\rho + 1$ to $\log(s+1) + i\rho$ are the columns of bundle $B_i$. In order to distinguish the rows in $I_1$ from the all-zero row, it is necessary, for each bundle $B_i$, to pick at least one column in the set of columns belonging to $B_i$ into the solution.

The second part, $A[I_2, \ast]$, of the rows of $A$ consists of $2s$ rows, that is $I_2 = [\log r + 2, \log r + 3, \ldots, \log r + 2s + 1]$. For each $i, j \in \mathbb{N}$ with $1 \leq i \leq 2^l - 1$ let $\text{bin}(i, j)$ be the binary $\{0, 1\}$-encoding of $i$ with exactly $j$ bits, padded with leading zeros if necessary. For each bundle $B_i$, fix an ordering of the at most $\rho$ truth assignments to variables in $B_i$. Recall that we may have repeated variables in $B_i$. If so, then repeat truth assignments in the order fixed above so that their overall number is exactly $\rho$. For each $p \in [\rho]$ and $q \in [s]$, let $\text{sat}_i(p, q) = 1$ if the $p$-th truth

---

(1) We note that the construction works as long as the number of bundles is $O(\log r)$ and each bundle’s size is $o(r)$. We opted for $\log r$ bundles as a natural choice.
A Double-Exponential Lower Bound for the Distinct Vectors Problem

assignment makes clause $C_q$ true and let $sat_i(p, q) = 0$ otherwise. Let $sat_i(\ast, q) = (sat_i(p, q))_{p \in \rho}$ and $sat(q) = sat_1(\ast, q) \circ sat_2(\ast, q) \circ \ldots \circ sat_{\log r}(\ast, q)$. Define the $(2q - 1)$-th row in $I_2$, $q \in [s]$, by

$$A[log r + 2q, \ast] = bin(q, log(s + 1)) \circ sat(q).$$

We call these rows the odd rows in $I_2$. Define the $2q$-th row in $I_2$, $q \in [s]$, by

$$A[log r + 2q + 1, \ast] = bin(q, log(s + 1)) \circ 0^{(n - log(s + 1))}.$$

These are the even rows in $I_2$. We say that the $(2q - 1)$-th and the $2q$-th rows correspond to clause $q$.

Finally, set $k = log(s + 1) + log r$. This concludes the construction of the DISTINCT VECTORS instance $(A, k)$.

Before proving the correctness, observe that all our other requirements on the construction are satisfied: For the number $k$ of columns to select, we have (recall that $s \leq Cr$)

$$k = log(s + 1) + log r \leq log(2s) + log r = log(2C) + 2 log r.$$

Moreover, number $n$ of columns satisfies $n = log(s + 1) + \rho log r = 2^{O(r/\log r)}$; and the number $m$ of rows satisfies $m = 1 + log r + 2s = O(r)$, each as required. Furthermore, since there are $2^{O(r/\log r)}$ truth assignments to the variables in each bundle, the reduction can be carried out in $2^{O(r/\log r)}$ time.

Correctness. We now prove that there is a solution to the above-constructed instance $(A, k)$ of DISTINCT VECTORS if and only if $\phi$ is satisfiable.

Assume that $(A, k)$ has a solution $K$. First, note that the even rows in $A[I_2, \ast]$ together with the all-zero row in $I_1$ are $s + 1$ rows that pairwise differ only in the first $log(s + 1)$ columns. Since for each $t \in \mathbb{N}$ we have that $t$ selected columns can pairwise distinguish at most $2^t$ rows, we thus have $[log(s + 1)] \subseteq K$. Let $K' = K \setminus [log(s + 1)]$ and observe $|K'| \leq log r$. Observe that in $A[I_1, \ast]$ there are $log r$ rows that each differ from the all-zero column in $A[I_1, \ast]$ only in the columns corresponding to some distinct bundle. Thus, for each bundle $B_j$, there is exactly one column, say $r_j$, in $K' \cap R_j$ where $R_j$ is the set of $B_j$’s columns, and no other columns are in $K'$. Observe that each $r_j$ corresponds by construction to a truth assignment to variables in $B_j$. Call this truth assignment $\alpha_j$. Thus, taking the union over all $i \in [log r]$ of the truth assignment $\alpha_i$ to the variables in $B_i$ represented by $r_i$, we get a truth assignment $\alpha$ to all variables in $X$. This truth assignment $\alpha$ is well-defined since the bundles constitute a partition of the variables. We claim that $\alpha$ satisfies $\phi$.

Since $K$ is a solution, for each $q \in [s]$, the sub-row $A[log r + 2q, K]$ is different from $A[log r + 2q + 1, K]$. These two sub-rows differ only in columns of bundles $B_j$ that correspond to some truth assignment to the variables in $B_j$ that satisfies clause $C_q$. Thus, $\alpha$ satisfies $C_q$ and indeed, since this holds for all $q \in [s]$, $\alpha$ satisfies $\phi$, as required.

Now assume that there is a truth assignment $\alpha$ to variables in $X$ that satisfies $\phi$. For each bundle $B_j$, there is a column $r_j$ in $B_j$’s columns such that the corresponding truth assignment, call it $\alpha_j$, assigns to variables in $B_i$ the same truth values as $\alpha$. We construct a solution $K$ to $(A, k)$ as follows. First, we put $[log(s + 1)] \subseteq K$. Then, for each bundle $B_j$ put $r_j \in K$. This concludes the construction. Observe that $|K| = log(s + 1) + log r$, as required. It remains to show that all rows in $A[\ast, K]$ are distinct.

Consider two rows $i, j \in [m]$, where $i \neq j$. We distinguish the following cases.

Case 1) $i, j \in I_1$. Then, one of the two rows, say $i$, has 1 in the columns of some bundle and row $j$ has 0 in these columns. Since by construction $K$ contains exactly one column from the columns of each bundle, thus, $A[i, K] \neq A[j, K]$.

Case 2) $i \in I_1$ and $j \in I_2$. Observe that each row in $I_1$ has only zeros in the first $log(s + 1)$ columns and each row in $I_2$ has at least one zero in the first $log(s + 1)$ columns. Thus, $A[i, K] \neq A[j, K]$.

Case 3) $i, j \in I_2$. If $A[i, K]$ and $A[j, K]$ differ in the first $log(s + 1)$ columns, then we are done. Otherwise, both $i$ and $j$ correspond to the same clause, say $C_q$, and they are not both even or both odd rows. Say $i$ is an odd and $j$ is an even row. By the definition of $K$, there is a bundle $B_i$ and a column $r_i$ such that $\alpha_i$ satisfies $C_q$. Thus, $A[i, r_i] = 1 \neq 0 = A[j, r_j]$ by construction of the two rows.

Thus, $K$ is a solution to $(A, k)$, as required. This concludes the proof.

3 Acknowledgments

We thank Vincent Froese and Irene Muzi for interesting and helpful discussions. We thank three anonymous reviewers that improved the presentation of the paper.
References

[BFS17] Thomas Bläsius, Tobias Friedrich, and Martin Schirneck. “The Parameterized Complexity of Dependency Detection in Relational Databases”. In: Proceedings of the 11th International Symposium on Parameterized and Exact Computation (IPEC’16). Vol. 63. LIPIcs. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, 6:1–6:13. DOI: 10.4230/LIPIcs.IPEC.2016.6.

[Cha+00] Moses Charikar, Venkatesan Guruswami, Ravi Kumar, Sridhar Rajagopalan, and Amit Sahai. “Combinatorial Feature Selection Problems”. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FoCS’00). 2000, pp. 631–640. DOI: 10.1109/SFCS.2000.892331.

[Fro+16] Vincent Froese, René van Bevern, Rolf Niedermeier, and Manuel Sorge. “Exploiting Hidden Structure in Selecting Dimensions That Distinguish Vectors”. In: Journal of Computer and System Sciences 82.3 (2016), pp. 521–535. DOI: 10.1016/j.jcss.2015.11.011.

[Fro18] Vincent Froese. “Fine-Grained Complexity Analysis of Some Combinatorial Data Science Problems”. PhD thesis. Technische Universität Berlin, 2018. DOI: 10.14279/depositonce-7123.

[IP01] Russell Impagliazzo and Ramamohan Paturi. “On the Complexity of k-SAT”. In: Journal of Computer and System Sciences 62.2 (2001), pp. 367–375. DOI: 10.1006/jcss.2000.1727.

[IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. “Which Problems Have Strongly Exponential Complexity?” In: Journal of Computer and System Sciences 63.4 (2001), pp. 512–530. DOI: 10.1006/jcss.2001.1774.

[Paw91] Zdzisław Pawlak. Rough sets - Theoretical Aspects of Reasoning about Data. Vol. 9. Theory and decision library : series D. Kluwer, 1991. DOI: 10.1007/978-94-011-3534-4.