

# Sparse Regression via Range Counting<sup>\*†</sup>

Jean Cardinal<sup>1</sup> and Aurélien Ooms<sup>2</sup>

1 Université libre de Bruxelles (ULB), Brussels, Belgium

jcardin@ulb.ac.be

2 BARC, University of Copenhagen, Denmark

aurelien.ooms@di.ku.dk

---

## Abstract

The sparse regression problem, also known as best subset selection problem, can be cast as follows: Given a set  $S$  of  $n$  points in  $\mathbb{R}^d$ , a point  $y \in \mathbb{R}^d$ , and an integer  $2 \leq k \leq d$ , find an affine combination of at most  $k$  points of  $S$  that is nearest to  $y$ . We describe a  $O(n^{k-1} \log^{d-k+2} n)$ -time randomized  $(1 + \varepsilon)$ -approximation algorithm for this problem with  $d$  and  $\varepsilon$  constant. This is the first algorithm for this problem running in time  $o(n^k)$ . Its running time is similar to the query time of a data structure recently proposed by Har-Peled, Indyk, and Mahabadi (ICALP'18), while not requiring any preprocessing. Up to polylogarithmic factors, it matches a conditional lower bound relying on a conjecture about affine degeneracy testing. In the special case where  $k = d = O(1)$ , we provide a simple  $O_\delta(n^{d-1+\delta})$ -time deterministic exact algorithm, for any  $\delta > 0$ . Finally, we show how to adapt the approximation algorithm for the sparse linear regression and sparse convex regression problems with the same running time, up to polylogarithmic factors.

## 1 Introduction

Searching for a point in a set that is the closest to a given query point is certainly among the most fundamental problems in computational geometry. It motivated the study of crucial concepts such as multidimensional search data structures, Voronoi diagrams, dimensionality reduction, and has immediate applications in the fields of databases and machine learning. A natural generalization of this problem is to search not only for a single nearest neighbor, but rather for the nearest *combination* of a bounded number of points. More precisely, given an integer  $k$  and a query point  $y$ , we may wish to find an affine combination of  $k$  points of the set that is the nearest to  $y$ , among all possible such combinations. This problem has a natural interpretation in terms of sparse approximate solutions to linear systems, and is known as the *sparse regression*, or *sparse approximation* problem in the statistics and machine learning literature. Sparsity is defined here in terms of the  $\ell_0$  pseudonorm  $\|\cdot\|_0$ , the number of nonzero components. The *sparse affine regression* problem can be cast as follows:

► **Problem 1 (Sparse affine regression).** Given a matrix  $A \in \mathbb{R}^{d \times n}$ , a vector  $y \in \mathbb{R}^d$ , and an integer  $2 \leq k \leq d$ , find  $x \in \mathbb{R}^n$  minimizing  $\|Ax - y\|_2$ , and such that  $\|x\|_0 \leq k$ , and  $\sum_{i=1}^n x_i = 1$ .

By interpreting the columns of  $A$  as a set of  $n$  points in  $\mathbb{R}^d$ , the problem can be reformulated in geometric terms as the *nearest induced flat* problem.

► **Problem 2 (Nearest induced flat).** Given a set  $S$  of  $n$  points in  $\mathbb{R}^d$ , an additional point  $y \in \mathbb{R}^d$ , and an integer  $k$  such that  $2 \leq k \leq d$ , find  $k$  points of  $S$  such that the distance from  $y$  to their affine hull is the smallest.

---

\* First author supported by the Fonds de la Recherche Scientifique-FNRS under CDR Grant J.0146.18. Second author supported by the VILLUM Foundation grant 16582. Part of this research was accomplished while the second author was a PhD student at ULB under FRIA Grant 5203818F (FNRS).

† A full version of this paper is available at <https://arxiv.org/abs/1908.00351> [8].

## 68:2 Sparse Regression via Range Counting

Here the distance from a point to a flat is the distance to the closest point on the flat. Note that if we allow  $k = 1$  in the definition above, we have the nearest neighbor problem as a special case. We consider the setting in which the dimension  $d$  of the ambient space as well as the number  $k$  of points in the sought combination are constant, and study the asymptotic complexity of the problem with respect to  $n$ . As observed recently by Har-Peled, Indyk, and Mahabadi [15], the problem is closely related to the classical *affine degeneracy testing* problem, defined as follows.

► **Problem 3 (Affine degeneracy testing).** Given a set  $S$  of  $n$  points in  $\mathbb{R}^d$ , decide whether there exist  $d + 1$  distinct points of  $S$  lying on an affine hyperplane.

The latter can be cast as deciding whether a point set is in so-called *general position*, as is often assumed in computational geometry problems. In the special case  $d = 2$ , the problem is known to be 3SUM-hard [14, 7]. In general, it is not known whether it can be solved in time  $O(n^{d-\delta})$  for some positive  $\delta$  [13, 2], even for randomized algorithms. Supposing it cannot, we directly obtain a conditional lower bound on the complexity of the nearest induced flat problem. This holds even for approximation algorithms, which return an induced flat whose distance is within some bounded factor of the distance of the actual nearest flat.

► **Lemma 1.1** (Har-Peled, Indyk, and Mahabadi [15]). *If the nearest induced flat problem can be approximated within any multiplicative factor in time  $O(n^{k-1-\delta})$  for some positive  $\delta$ , then affine degeneracy testing can be solved in time  $O(n^{d-\delta})$ .*

**Proof.** Suppose we have an approximation algorithm for the nearest induced flat problem. Then given an instance of affine degeneracy testing, we can go through every point  $y \in S$  and run this algorithm on an instance composed of the set  $S \setminus \{y\}$ , the point  $y$ , and  $k = d$ . The answer to the degeneracy testing instance is positive if and only if for at least one of these instances, the distance to the approximate nearest flat is zero. The running time is  $O(n^{d-\delta})$ . ◀

For more motivation on the subject, see the full version [8].

### Our results

We prove that the nearest induced flat problem (Problem 2), can be solved within a  $(1 + \varepsilon)$  approximation factor for constant  $d$  and  $\varepsilon$  in time  $O(n^{k-1} \log^{d-k+2} n)$ , which matches the conditional lower bound on affine degeneracy testing, up to polylogarithmic factors. Har-Peled, Indyk, and Mahabadi [15] gave a data structure to preprocess a set of data points to allow solving the nearest induced flat problem on this set for any query point. Their data structure requires  $\tilde{O}(n^k)$  preprocessing and  $\tilde{O}(n^{k-1})$  query time. We propose an algorithm that gets rid of the preprocessing for single queries: the overall running time of our algorithm is equal to the query time of their data structure, up to polylogarithmic factors. To the best of our knowledge, this is a near-linear improvement on all previous methods for this special case.

The two main tools that are used in our algorithms are on the one hand the approximation of the Euclidean distance by a polyhedral distance, as is done in Agarwal, Rubin, and Sharir [1], and on the other hand a reduction of the decision version of the problem to orthogonal range queries. Note that orthogonal range searching data structures are also used in [15], albeit in a significantly distinct fashion.

In §2, as warm-up, we focus on the special case of Problem 2 in which  $d = 3$  and  $k = 2$ .

■ **Table 1** Results. For the approximation algorithms, the dependency on  $\varepsilon$  in the running time is of the order of  $\varepsilon^{(1-d)/2}$ . Only the details for the first result are included in this abstract. The details for the other results can be found in the full version [8].

Problem	Details	Approximation	Running Time
Problem 4: Nearest induced line in $\mathbb{R}^3$	§2	$1 + \varepsilon$	$O_\varepsilon(n \log^3 n)$
Problem 2: Nearest induced flat	[8]	$1 + \varepsilon$	$O_{d,\varepsilon}(n^{k-1} \log^{d-k+2} n)$
Problem 5: Nearest induced hyperplane	[8]	1	$O_{d,\delta}(n^{d-1+\delta}), \forall \delta > 0$
Problem 6: Nearest induced simplex	[8]	$1 + \varepsilon$	$O_{d,\varepsilon}(n^{k-1} \log^d n)$

► **Problem 4** (Nearest induced line in  $\mathbb{R}^3$ ). Given a set  $S$  of  $n$  points in  $\mathbb{R}^3$ , and an additional point  $y$ , find two points  $a, b \in S$  such that the distance from  $y$  to the line going through  $a$  and  $b$  is the smallest.

Our algorithm for this special case already uses all the tools that are subsequently generalized for arbitrary values of  $k$  and  $d$ . The general algorithm for the nearest induced flat problem is described in the full version [8]. In the full version, we also consider the special case of Problem 2 in which  $k = d$ , which can be cast as the *nearest induced hyperplane* problem.

► **Problem 5** (Nearest induced hyperplane). Given a set  $S$  of  $n$  points in  $\mathbb{R}^d$ , and an additional point  $y$ , find  $d$  points of  $S$  such that the distance from  $y$  to the affine hyperplane spanned by the  $d$  points is the smallest.

For this case, we design an *exact* algorithm with running time  $O(n^{d-1+\delta})$ , for any  $\delta > 0$ . The solution solely relies on classical computational geometry tools, namely point-hyperplane duality and cuttings [11, 10].

Our algorithms can be adapted to perform *sparse linear regression*, instead of sparse affine regression. In the former, we drop the condition that the sum of the coefficients must be equal to one. This is equivalent to the nearest *linear* induced  $k$ -flat problem. It can be solved in the same time as in the affine case. To see this, realize that the problem is similar to the nearest induced flat problem where the first vertex is always the origin. The obtained complexity is the same as the one for the nearest induced flat problem.

Adapting our algorithm to sparse *convex* regression, which differs from sparse affine regression by requiring  $x$  to be positive, is a bit more involved. The sparse convex regression problem can be cast as the problem of finding the nearest simplex induced by  $k$  points of  $S$ .

► **Problem 6** (Nearest induced simplex). Given a set  $S$  of  $n$  points in  $\mathbb{R}^d$ , an additional point  $y$ , and an integer  $k$  such that  $2 \leq k \leq d$ , find  $k$  points of  $S$  such that the distance from  $y$  to their convex hull is the smallest.

We prove that this problem can also be approximated within a  $(1 + \varepsilon)$  approximation factor for constant  $d$  and  $\varepsilon$  in time  $O(n^{k-1} \log^d n)$ , hence with an extra  $O(\log^{k-2} n)$  factor in the running time compared to the affine case. This is described in the full version [8].

Our results and the corresponding sections are summarized in Table 1.

## 2 A $(1 + \varepsilon)$ -approximation algorithm for the nearest induced line problem in $\mathbb{R}^3$

We first consider the nearest induced line problem (Problem 4). We describe a near-linear time algorithm that returns a  $(1 + \varepsilon)$ -approximation to the nearest induced line in  $\mathbb{R}^3$ , that is, a line at distance at most  $(1 + \varepsilon)$  times larger than the distance to the nearest line.

## 68:4 Sparse Regression via Range Counting

► **Theorem 2.1.** *For any constant  $\varepsilon > 0$ , there is a randomized  $(1 + \varepsilon)$ -approximation algorithm for the nearest induced line problem in  $\mathbb{R}^3$  running in time  $O_\varepsilon(n \log^3 n)$  with high probability.*

The sketch of our algorithm is as follows: First, reduce the problem of minimizing the Euclidean distance to that of minimizing the polyhedral distance for some well-chosen polyhedron depending on  $\varepsilon$ . Second, reduce the problem of minimizing the polyhedral distance to that of edge-shooting. Third, reduce the problem of edge-shooting to that of deciding whether an edge shot at a certain distance would hit any induced line through some sort of binary search. Fourth, efficiently solve this decision problem using orthogonal range counting data structures.

### $(1 + \varepsilon)$ -approximation via polyhedral distances.

The *polyhedral distance*  $d_Q(y, v)$  between two points  $y$  and  $v$  with respect to a polyhedron  $Q$  centered on the origin is the smallest  $\lambda$  such that the dilation  $\lambda Q$  of  $Q$  centered on  $y$  contains  $v$ , hence such that  $v \in y + \lambda Q$ . Our proof uses the following result, of which a weaker variant due to Dudley [12] is a major ingredient in the design of the data structure described by Agarwal, Rubin, and Sharir [1].

► **Lemma 2.2** (Arya, Arya, da Fonseca, Mount [3]). *For any positive integer  $d$  and positive real  $\varepsilon$ , there exists a  $d$ -dimensional polyhedron  $Q$  with  $O(1/\varepsilon^{(d-1)/2})$  faces such that for every  $y, v \in \mathbb{R}^d$ :*

$$\|y - v\|_2 \leq d_Q(y, v) \leq (1 + \varepsilon) \cdot \|y - v\|_2.$$

This bound is asymptotically optimal. See [4, 6, 5] for more details.

Next, we reduce Problem 4 to a counting problem in two steps.

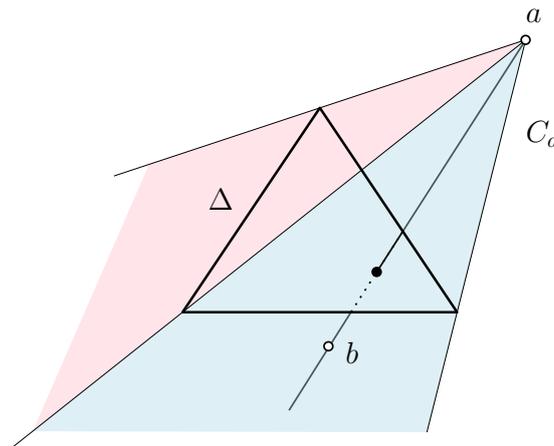
### Edge-shooting.

We use Lemma 2.2 for  $d = 3$ . We give an exact algorithm for computing the nearest induced line with respect to a polyhedral distance  $d_Q$ , where  $Q$  is defined from  $\varepsilon$  as in Lemma 2.2. Given a polyhedron  $Q$ , one can turn it into a simplicial polyhedron by triangulating it. Therefore, for constant values of  $\varepsilon$ , this reduces the problem to a constant number of instances of the *edge-shooting problem*, defined as follows: Given an edge  $e$  of  $Q$ , find the smallest value  $\lambda$  such that  $y + \lambda e$  intersects a line through two points of  $S$ . We iterate this for all edges of  $Q$ , and pick the minimum value. This is exactly the polyhedral distance from  $y$  to its nearest induced line.

### Binary search.

Using a randomized binary search procedure, we reduce the edge-shooting problem to a *counting problem*, defined as follows: given the triangle  $\Delta$  defined as the convex hull of  $y$  and  $y + \lambda e$ , count how many pairs of points  $a, b \in S$  are such that the line  $\ell(a, b)$  through them intersects  $\Delta$ . Suppose there exists a procedure for solving this problem. We can use this procedure to solve the edge-shooting problem efficiently as follows.

First initialize  $\lambda$  to some upper bound on the distance. Then count how many lines  $\ell(a, b)$  intersect  $\Delta$ , using the procedure. If there is only one, then return its (polyhedral) distance to  $y$ . Otherwise, pick one such line uniformly at random and compute the value  $\lambda'$  such that this line intersects  $y + \lambda'e$ . Then iterate the previous steps with  $\lambda \leftarrow \lambda'$ , unless  $\lambda' = 0$  in which case we return 0. Since we picked the line at random, and since there are  $O(n^2)$  such



■ **Figure 1** The cone  $C_a$ .

lines at the beginning of the search, the number of iterations of this binary search is  $O(\log n)$  with high probability.

We therefore reduced the nearest induced line problem to  $O(\varepsilon^{-1} \log n)$  instances of the counting problem.

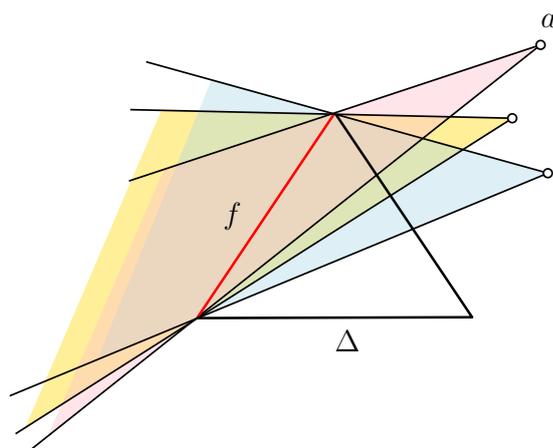
**Orthogonal range counting queries.**

Data structures for *orthogonal range counting queries* store a set of points in  $\mathbb{R}^g$  in such a way that the number of points in a given  $g$ -rectangle (cartesian product of  $g$  intervals) can be returned quickly. Known data structures for orthogonal range counting queries in  $\mathbb{R}^g$  require  $O(n \log^{g-1} n)$  preprocessing time and can answer queries in  $O(\log^{g-1} n)$  time [16, 9]. Note that the actual coordinates of the points do not matter: We only need to know the order of their projections on each axis. We now show how to solve the counting problem using a data structure for orthogonal range queries in  $\mathbb{R}^3$ .

Let us fix the triangle  $\Delta$  and a point  $a \in \mathbb{R}^3$ , and consider the locus of points  $b \in \mathbb{R}^3$  such that the line  $\ell(a, b)$  intersects  $\Delta$ . This is a double simplicial cone with apex  $a$  and whose boundary contains the boundary of  $\Delta$ . This double cone is bounded by three planes, one for each edge of  $\Delta$ . In fact, we will only consider one of the two cones, because  $\ell(a, b)$  intersects  $\Delta$  if and only if either  $b$  is contained in the cone of apex  $a$ , or  $a$  is contained in the cone of apex  $b$ . Let us call  $C_a$  the cone of apex  $a$ . This is illustrated on Figure 1.

Let us consider one edge  $f$  of  $\Delta$  and all the planes containing  $f$ . These planes induce a circular order on the points of  $S$ , which is the order in which they are met by a plane rotating around the supporting line of  $f$ . This is illustrated on Figure 2. Now let us denote by  $H_f$  the plane containing  $a$  and  $f$  and by  $H_f^+$  the halfspace bounded by  $H_f$  and containing  $\Delta$ . The set of points of  $S$  contained in  $H_f^+$  is an interval in the circular order mentioned above. Hence the set of points contained in  $C_a$  is the intersection of three intervals in the three circular orders defined by the three edges of  $\Delta$ .

**Proof of Theorem 2.1.** Let  $Q$  be some polyhedron in  $\mathbb{R}^3$ ,  $\lambda \in \mathbb{R}$ ,  $S \subset \mathbb{R}^3$ ,  $y \in \mathbb{R}^3$ , and  $e$  an edge of  $Q$ . We use an orthogonal range counting data structure for storing the points of  $S$  with coordinates corresponding to their ranks in each of the three permutations induced by the three edges of  $\Delta = \text{conv}(\{y, y + \lambda e\})$ . We get those rank-coordinates by sorting  $S$  three



■ **Figure 2** The order of the points defined by the planes containing an edge  $f$  of  $\Delta$ .

times, once for each induced permutation, in time  $O(n \log n)$ , then construct the orthogonal range counting data structure with those coordinates in time  $O(n \log^2 n)$ . Then for each of the  $n$  points  $a \in S$ , we count the number of points  $b$  in the cone  $C_a$  by querying the data structure in  $O(\log^2 n)$  time. Hence overall, the counting problem is solved in time  $O(n \log^2 n)$ . Note that the circularity of the order can be easily handled by doubling every point.

This can be combined with the previous reductions provided we can choose a line intersecting  $\Delta$  uniformly at random within that time bound. This is achieved by first choosing  $a$  with probability proportional to the number of points  $b$  such that  $\ell(a, b) \cap \Delta \neq \emptyset$ . Then we can pick a point  $b$  uniformly at random in this set in linear time.

Combining with the previous reductions, we obtain an approximation algorithm running in time  $O_\epsilon(n \log^3 n)$  for the nearest induced line problem in  $\mathbb{R}^3$ . ◀

---

## References

- 1 Pankaj K. Agarwal, Natan Rubin, and Micha Sharir. Approximate nearest neighbor search amid higher-dimensional flats. In *25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria*, pages 4:1–4:13, 2017. doi:10.4230/LIPIcs.ESA.2017.4.
- 2 Nir Ailon and Bernard Chazelle. Lower bounds for linear degeneracy testing. *J. ACM*, 52(2):157–171, 2005. doi:10.1145/1059513.1059515.
- 3 Rahul Arya, Sunil Arya, Guilherme D. da Fonseca, and David M. Mount. Optimal bound on the combinatorial complexity of approximating polytopes. In *SODA*, pages 786–805. SIAM, 2020.
- 4 Sunil Arya, Guilherme D. da Fonseca, and David M. Mount. On the combinatorial complexity of approximating polytopes, 4 2016. arXiv:1604.01175v4.
- 5 Sunil Arya, Guilherme D. da Fonseca, and David M. Mount. Approximate convex intersection detection with applications to width and minkowski sums, 7 2018. arXiv:1807.00484v1.
- 6 Sunil Arya, Guilherme Dias da Fonseca, and David M. Mount. Near-optimal epsilon-kernel construction and related problems. In *Symposium on Computational Geometry*, volume 77 of *LIPIcs*, pages 10:1–10:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

- 7 Luis Barba, Jean Cardinal, John Iacono, Stefan Langerman, Aurélien Ooms, and Noam Solomon. Subquadratic algorithms for algebraic 3SUM. *Discrete & Computational Geometry*, 61(4):698–734, 2019. doi:10.1007/s00454-018-0040-y.
- 8 Jean Cardinal and Aurélien Ooms. Sparse regression via range counting, 8 2019. arXiv:1908.00351.
- 9 Bernard Chazelle. A functional approach to data structures and its use in multidimensional searching. *SIAM J. Comput.*, 17(3):427–462, 1988. doi:10.1137/0217026.
- 10 Bernard Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete & Computational Geometry*, 9:145–158, 1993. doi:10.1007/BF02189314.
- 11 Bernard Chazelle and Joel Friedman. A deterministic view of random sampling and its use in geometry. *Combinatorica*, 10(3):229–249, 1990. doi:10.1007/BF02122778.
- 12 Richard M. Dudley. Metric entropy of some classes of sets with differentiable boundaries. *Journal of Approximation Theory*, 10(3):227 – 236, 1974. URL: <http://www.sciencedirect.com/science/article/pii/0021904574901208>, doi:[https://doi.org/10.1016/0021-9045\(74\)90120-8](https://doi.org/10.1016/0021-9045(74)90120-8).
- 13 Jeff Erickson and Raimund Seidel. Better lower bounds on detecting affine and spherical degeneracies. *Discrete & Computational Geometry*, 13:41–57, 1995. doi:10.1007/BF02574027.
- 14 Anka Gajentaan and Mark H. Overmars. On a class of  $o(n^2)$  problems in computational geometry. *Comput. Geom.*, 5:165–185, 1995. doi:10.1016/0925-7721(95)00022-2.
- 15 Sariel Har-Peled, Piotr Indyk, and Sepideh Mahabadi. Approximate sparse linear regression. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, pages 77:1–77:14, 2018. doi:10.4230/LIPIcs.ICALP.2018.77.
- 16 Dan E. Willard. New data structures for orthogonal range queries. *SIAM J. Comput.*, 14(1):232–253, 1985.