COMPRESSED LEARNING OF HIGH-DIMENSIONAL SPARSE FUNCTIONS

Karin Schnass, Jan Vybíral

Johann Radon Institute for Computational and Applied Mathematics
Austrian Academy of Sciences, Altenbergerstraße 69, A-4040 Linz, Austria

ABSTRACT

This paper presents a simple randomised algorithm for recovering high-dimensional sparse functions, i.e. functions $f : [0,1]^d \to \mathbb{R}$ which depend effectively only on k out of d variables, meaning $f(x_1,\dots,x_d) = g(x_{i_1},\dots,x_{i_k})$, where the indices $1 \le i_1 < i_2 < \cdots < i_k \le d$ are unknown. It is shown that (under certain conditions on g) this algorithm recovers the k unknown coordinates with probability at least $1-6\exp(-L)$ using only $\mathcal{O}(k(L+\log k)(L+\log d))$ samples of f.

Index Terms— High dimensional function approximation, random algorithm, Hoeffding's inequality, concentration of measure

1. INTRODUCTION

Assume, that we want to approximate a function $f : [0,1]^d \to \mathbb{R}$ using only a small number of its function values. This is for instance the case when the function describes a physical process and every evaluation corresponds to running a large scale experiment. Of course, the more precisely we want to recover f, the more samples of f we have to take. It is a well known fact [1, 2], that the number of samples needed to reach a given precision $\varepsilon > 0$ grows exponentially with d even for $\mathcal{C}^\infty$ functions.

Therefore, to reach a better result, we have to restrict ourselves to the cases, where f enjoys some special structure. In this short note we study the case when $f : [0,1]^d \to \mathbb{R}$ depends effectively only on $k \ll d$ variables, i.e.

$$f(x) = f(x_1,\dots,x_d) = g(x_{i_1},\dots,x_{i_k}) = g(x_I). \qquad (1)$$

Here the set $I = \{i_1,\dots,i_k\} \subseteq \{1,\dots,d\}$ collects the k (unknown) active coordinates i_ℓ and g is a twice continuously differentiable function.

Obviously, the problem consists of two parts. First, one has to locate the effective coordinates, $i \in I$. Then one has to approximate the function $g : [0,1]^k \to \mathbb{R}$. This paper gives a probabilistic algorithm which, under certain conditions on g, answers the first part of this problem with high probability and using only a relatively small number of samples. The second part may then be handled by standard techniques of approximation theory and we will not go into much detail on that.

First, let us give a brief overview of known results. Functions of type (1) were recently studied using deterministic algorithms in [3]. In particular, the authors of [3] describe, how to approximate f uniformly to accuracy $\|g\|_{\mathrm{Lip}} h$ by evaluating the function on $2(k+1)e^{k+1}h^{-k}\log_2 d$ adaptively chosen points. Here, $h > 0$ is the chosen precision and g is assumed to be Lipschitz with its Lipschitz norm denoted by $\|g\|_{\mathrm{Lip}}$. Furthermore, (1) is a special case of

$$f(x) = g(Ax),$$

where A is a fixed (unknown) $k \times d$ matrix. This case was studied in [4] for $k = 1$ and in [5] for arbitrary $k < d$. The methods used there rely essentially on techniques from Compressed Sensing. Here we give an alternative approach based on several (rather elementary) concentration inequalities for random variables.

2. ALGORITHM

Let us first give a short sketch of the idea and outline the necessary ingredients for the main result. Similarly to the approach described in [4, 5], we rely on numerical approximations of directional derivatives $\frac{\partial f}{\partial \varphi}(x)$. For this reason, we assume, that f is actually defined on a small neighbourhood of $[0,1]^d$, namely on $D = (-\bar\epsilon, 1+\bar\epsilon)^d$. Let A denote the $k \times d$ matrix

$$A = \begin{pmatrix} e_{i_1}^T \\ \vdots \\ e_{i_k}^T \end{pmatrix},$$

where e_{i_j} are the canonical vectors[1] in $\mathbb{R}^d$. For $x \in [0,1]^d$, $\varphi \in \mathbb{R}^d$ with $\|\varphi\|_\infty := \max_i |\varphi_i| \le r$ and $\epsilon, r \in \mathbb{R}_+$, with

Both authors acknowledge the financial support provided by the START-award "Sparse Approximation and Optimization in High Dimensions" of the Fonds zur Förderung der wissenschaftlichen Forschung (FWF, Austrian Science Foundation).

[1] Here and in the remainder of the paper all vectors will be considered column vectors.

$r\epsilon < \bar{\epsilon}$, we get by Taylor expansion the identity

$$\nabla g(Ax)^T A\varphi = \frac{\partial f}{\partial \varphi}(x) = \frac{f(x+\epsilon\varphi) - f(x)}{\epsilon} - \frac{\epsilon}{2}[\varphi^T \nabla^2 f(\zeta)\varphi] \quad (2)$$

for a suitable $\zeta(x,\varphi) \in D$. We apply (2) to the set of points $\mathcal{X} = \{x^j \in [0,1]^d : j = 1,\dots,m_X\}$ drawn uniformly at random with respect to the Lebesgue measure and the set of directions $\Phi = \{\varphi^j \in \mathbb{R}^d, j = 1,\dots,m_\Phi\}$, where

$$\varphi_\ell^j = \begin{cases} 1/\sqrt{m_\Phi} & \text{with prob.} \quad 1/2 \\ -1/\sqrt{m_\Phi} & \text{with prob.} \quad 1/2 \end{cases},$$

for every $j \in \{1,\dots,m_\Phi\}$ and every $\ell \in \{1,\dots,d\}$. Actually we identify Φ with the $m_\Phi \times d$ matrix whose rows are the vectors $(\varphi^i)^T$. We rewrite the $m_X \times m_\Phi$ instances of (2) in matrix notation as

$$\Phi X = Y + \mathcal{E}, \quad (3)$$

where Y and $\mathcal{E}$ are the $m_\Phi \times m_X$ matrices defined entry-wise by

$$y_{ij} = \frac{f(x^j + \epsilon\varphi^i) - f(x^j)}{\epsilon}, \quad (4)$$

$$\varepsilon_{ij} = -\frac{\epsilon}{2}[(\varphi^i)^T \nabla^2 f(\zeta_{ij})\varphi^i], \quad (5)$$

and X is the $d \times m_X$ matrix with i-th row

$$X^i := \left(\frac{\partial g}{\partial z_i}(Ax^1), \dots, \frac{\partial g}{\partial z_i}(Ax^{m_X})\right),$$

for $i \in I$ and all other rows equal to zero. In the remainder we will also write shortly $\partial_i g$ for $\frac{\partial g}{\partial z_i}$.
Now we can already describe the idea, how to recover the (unknown) indices $i \in I$. The discussion above shows, that it is enough to identify the non zero rows of X. Multiplying (3) with Φ^T from the left-hand side, we get

$$\Phi^T \Phi X = \Phi^T Y + \Phi^T \mathcal{E}. \quad (6)$$

This identity is crucial for our algorithm. Observe, that Y is obtained by sampling f as described by (4), using $2m_X m_\Phi$ function evaluations, and $\Phi^T Y$ can be calculated by a matrix product. Looking at the random construction of $\Phi^T\Phi$ we see that in expectation it is identical to the $d \times d$ identity matrix. Thus we can expect it to behave essentially like that when applied to the rank k matrix X, i.e. $\Phi^T\Phi X \approx X$. Finally, $\Phi^T\mathcal{E}$ should be small as long as ϵ was chosen small enough, leading to $\Phi^T Y \approx \Phi^T\Phi X$. Putting these pieces together we get that

$$\Phi^T Y \approx X,$$

meaning that to identify the active components of f, we just need to select the k largest rows of $\Phi^T Y$ in the maximum norm.[2] To turn the sketch above into a mathematically sound statement we need to keep track of the following probabilities,

1. the probability that for every active coordinate the corresponding row in X has a certain size,
2. the probability that the indices of the k largest rows of $\Phi^T\Phi X$ in the maximum norm are the same as those of X,
3. the probability that the indices of the k largest rows of $\Phi^T Y = \Phi^T\Phi X - \Phi^T\mathcal{E}$ are the same as those of $\Phi^T\Phi X$.

The estimates of these three probabilities make heavy use of concentration properties of the random variables involved so far and form the heart of the proof of the following Theorem.

Theorem 1 *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a sparse function as described in* (1)*, that is defined and twice continuously differentiable on a small neighbourhood of $[0,1]^d$. For $L \le d$, a positive real number, the randomised algorithm described above recovers the k unknown active coordinates of f with probability at least $1 - 6\exp(-L)$ using only*

$$\mathcal{O}(k(L + \log k)(L + \log d)) \quad (7)$$

samples of f.

Note, that the constants involved in the $\mathcal{O}$ notation in (7) depend on smoothness properties of g, namely on the ratio of C_1/α, where

$$\alpha := \min_{i\in I} \|\partial_i g\|_1 \quad \text{and} \quad C_1 := \max_{i\in I} \|\partial_i g\|_\infty.$$

We postpone a detailed discussion of the result to Section 4 and now give the quite simple and intuitive proof.

3. PROOF

We start by estimating the first probability that for every active coordinate the maximum norm of the corresponding row X^i is of a certain size. To do this for $i \in I$ we will bound the maximum entry of the i-th row $\left(\frac{\partial g}{\partial z_i}(Ax^1), \dots, \frac{\partial g}{\partial z_i}(Ax^{m_X})\right)$ by its average and use Hoeffding's inequality, which we recall briefly below.

Proposition 1 (Hoeffding's inequality) *Let $Z_1,\dots,Z_m$ be independent random variables. Assume that the Z_i are almost surely bounded, i.e., there exist finite scalars a_j, b_j such that*

$$\mathbb{P}(Z_j \in [a_j, b_j]) = 1,$$

for $j = 1,\dots,m$. Then we have

$$\mathbb{P}\Big(\big|\sum_{j=1}^m (Z_j - \mathbb{E}Z_j)\big| \ge t\Big) \le 2\exp\left(-\frac{2t^2}{\sum_{j=1}^m (b_j - a_j)^2}\right).$$

[2] We expect the Euclidean norm to give even better results. However, here we selected the maximum norm, which allows for a short proof due to Lemma 1.

If we set $Z_j = |\frac{\partial g}{\partial z_i}(Ax^j)|$, we have

$$\begin{aligned}\mathbb{E}Z_j &= \int_{[0,1]^d} |\frac{\partial g}{\partial z_i}(Ax)|dx \\ &= \int_{[0,1]^k} |\frac{\partial g}{\partial z_i}(x)|dx := \|\partial_i g\|_1,\end{aligned}$$

and

$$0 \leq Z_j \leq \sup_{x\in[0,1]^k} |\frac{\partial g}{\partial z_i}(x)| := \|\partial_i g\|_\infty.$$

This leads to

$$\begin{aligned}\mathbb{P}\Big(|\sum_{j=1}^{m_X} |\frac{\partial g}{\partial z_i}(Ax^j)| - m_X\|\partial_i g\|_1| \geq t\Big) \\ \leq 2\exp\left(-\frac{2t^2}{m_X\|\partial_i g\|_\infty^2}\right),\end{aligned}$$

and after setting $t = s_1 m_X \|\partial_i g\|_1$ for $s_1 \in (0,1)$ to

$$\begin{aligned}\mathbb{P}\Big(\frac{1}{m_X}\sum_{j=1}^{m_X} |\frac{\partial g}{\partial z_i}(Ax^j)| \leq (1-s_1)\|\partial_i g\|_1\Big) \\ \leq 2\exp\left(-\frac{2m_X s_1^2\|\partial_i g\|_1^2}{\|\partial_i g\|_\infty^2}\right).\end{aligned}$$

Since $\|X^i\|_\infty = \max_j |X_{ij}| \geq \frac{1}{m_X}\sum_{j=1}^{m_X}|X_{ij}|$ we get the following estimate for the maximum norm of the row X^i corresponding to the active coordinate $i \in I$.

$$\begin{aligned}\mathbb{P}(\|X^i\|_\infty \leq (1-s_1)\|\partial_i g\|_1) \\ \leq 2\exp\left(-\frac{2m_X s_1^2\|\partial_i g\|_1^2}{\|\partial_i g\|_\infty^2}\right).\end{aligned}$$

Defining $\alpha = \min_{i\in I}\|\partial_i g\|_1$ and $C_1 = \max_{i\in I}\|\partial_i g\|_\infty$ this finally leads to

$$\begin{aligned}\mathbb{P}(\min_{i\in I}\|X^i\|_\infty \leq (1-s_1)\alpha) \\ \leq 2k\exp\left(-\frac{2m_X s_1^2\alpha^2}{C_1^2}\right) := p_1. \quad (8)\end{aligned}$$

Next we investigate the probability that the k largest rows of $\Phi^T\Phi X$ in the maximum norm are the same as those of X. To do this we will show that the magnitude of the entries remains roughly the same, using the following result from [6][3].

Lemma 1 ([6], Lemma III.1) *Let $x, y \in \mathbb{R}^d$ with $x, y \neq 0$. Assume that Φ is an $m_\Phi \times d$ random matrix with independent $\pm 1/\sqrt{m_\Phi}$ Bernoulli entries. Then for all $t > 0$*

$$\mathbb{P}\big(|\langle\Phi x, \Phi y\rangle - \langle x, y\rangle| \geq t\|x\|_2\|y\|_2\big) \leq 2\exp\left(-\frac{m_\Phi t^2}{3+4t}\right).$$

[3]We corrected and simplified the constants found therein.

From the observation that $(\Phi^T\Phi X)_{ij} = \langle\Phi e_i, \Phi X_j\rangle$, where X_j denotes the j-th column of X, and $X_{ij} = \langle e_i, X_j\rangle$ we get

$$\begin{aligned}\mathbb{P}\big(|(\Phi^T\Phi X)_{ij} - X_{ij}| \geq t\big) \\ = \mathbb{P}\big(|\langle\Phi e_i, \Phi X_j\rangle - \langle e_i, X_j\rangle| \geq t\big) \\ \leq 2\exp\left(-\frac{m_\Phi t^2}{3\|X_j\|_2^2 + 4t\|X_j\|_2}\right) \\ \leq 2\exp\left(-\frac{m_\Phi t^2}{3kC_1^2 + 4t\sqrt{k}C_1}\right),\end{aligned}$$

where for the last bound we used that

$$\|X_j\|_2^2 = \sum_{i\in I}|\frac{\partial g}{\partial z_i}(Ax^j)|^2 \leq \sum_{i\in I}\|\partial_i g\|_\infty^2 \leq kC_1^2.$$

Setting $t = s_2\alpha$ leads to

$$\begin{aligned}\mathbb{P}\big(\max_{i,j}|(\Phi^T\Phi X)_{ij} - X_{ij}| \geq s_2\alpha\big) \\ \leq 2dm_X\exp\left(-\frac{m_\Phi s_2^2\alpha^2}{3kC_1^2 + 4s_2\alpha\sqrt{k}C_1}\right),\end{aligned}$$

which can be further simplified to

$$\begin{aligned}\mathbb{P}\big(\max_{i,j}|(\Phi^T\Phi X)_{ij} - X_{ij}| \geq s_2\alpha\big) \\ \leq 2dm_X\exp\left(-\frac{m_\Phi s_2^2\alpha^2}{6kC_1^2}\right) := p_2, \quad (9)\end{aligned}$$

as long as s_2 is chosen smaller than $3/4$.
Finally we estimate the third probability that the k largest rows of $\Phi^T Y = \Phi^T\Phi X - \Phi^T\mathcal{E}$ are the same as those of $\Phi^T\Phi X$ by showing that the entries of $\Phi^T\mathcal{E}$ are very likely to be small. The ij-th entry of the matrix $\Phi^T\mathcal{E}$ can be written as

$$(\Phi^T\mathcal{E})_{ij} = \sum_{\ell=1}^{m_\Phi}\varphi_i^\ell\varepsilon_{\ell j}.$$

Thus setting $Z_\ell = \varphi_i^\ell\varepsilon_{\ell j}$ and observing that Z_ℓ takes only the values $\pm\varepsilon_{\ell j}/\sqrt{m_\phi}$, we can use again Hoeffding's inequality to get

$$\mathbb{P}(|(\Phi^T\mathcal{E})_{ij}| \geq t) \leq 2\exp\left(-\frac{m_\Phi t^2}{2\sum_{\ell=1}^{m_\Phi}\varepsilon_{\ell j}^2}\right).$$

From Equation (5) we can bound the entries of $\mathcal{E}$ by

$$\begin{aligned}|\varepsilon_{ij}| &= \frac{\epsilon}{2}|(\varphi^i)^T\nabla^2 f(\zeta_{ij})\varphi^i| \\ &= \frac{\epsilon}{2}\Big|\sum_{\ell,\ell'=1}^{d}\varphi_\ell^i[\partial_\ell\partial_{\ell'}f(\zeta_{ij})]\varphi_{\ell'}^i\Big| \\ &= \frac{\epsilon}{2}\Big|\sum_{\ell,\ell'\in I}\varphi_\ell^i[\partial_\ell\partial_{\ell'}g(A\zeta_{ij})]\varphi_{\ell'}^i\Big| \\ &\leq \frac{\epsilon k^2}{2m_\phi}\max_{\ell,\ell'\in I}\|\partial_\ell\partial_{\ell'}g\|_\infty := \frac{\epsilon k^2}{2m_\phi}C_2.\end{aligned}$$

Using this estimate to bound $\sum_{\ell=1}^{m_\Phi} \varepsilon_{\ell j}^2$ we arrive at

$$\mathbb{P}(|(\Phi^T \mathcal{E})_{ij}| \geq t) \leq 2 \exp\left(-\frac{2m_\Phi^2 t^2}{\epsilon^2 k^4 C_2^2}\right),$$

and setting $t = s_3\alpha$ at

$$\mathbb{P}(\max_{ij} |(\Phi^T \mathcal{E})_{ij}| \geq s_3\alpha) \leq 2dm_X \exp\left(-\frac{2m_\Phi^2 s_3^2 \alpha^2}{\epsilon^2 k^4 C_2^2}\right) := p_3. \quad (10)$$

Combining the estimates in (8-10) we see that with high probability the rows of $\Phi^T Y$ corresponding to the active coordinates have maximum norm of at least $\alpha(1 - s_1 - s_2 - s_3)$, while the rows of $\Phi^T Y$ corresponding to the inactive coordinates have maximum norm of at most $\alpha(s_2 + s_3)$. Thus as long as

$$\alpha(1 - s_1 - s_2 - s_3) > \alpha(s_2 + s_3)$$

or

$$s_1 + 2s_2 + 2s_3 < 1$$

our strategy will work with high probability, namely at least with probability

$$1 - p_1 - p_2 - p_3,$$

with p_i as defined in (8-10). We want to minimise $p_1 + p_2 + p_3$ and the product $m_\Phi \times m_X$ while keeping $\epsilon > 0$ as large as possible. Setting $s_1 = s_2 = s_3 = 1/6$ and $p_1 = p_2 = p_3 = 2\exp(-L)$ for $0 < L \leq d$, we obtain

$$m_X = \frac{18C_1^2(L + \log k)}{\alpha^2},$$

using (8) and

$$m_\Phi = (L + \log(dm_X))\frac{216kC_1^2}{\alpha^2},$$

using (9). Finally, to get $p_3 \leq p_2$, we set in (10)

$$\epsilon^2 := \min\left(\frac{12m_\Phi C_1^2}{k^3 C_2^2}, m_\Phi \bar{\epsilon}^2\right). \quad (11)$$

Thus, to reach the probability of success $1 - 6\exp(-L)$, we need

$$m_\Phi \times m_X \approx k(L + \log k)(L + \log[d(L + \log k)])$$

samples. As $L + \log k \lesssim d$, this can be simplified to

$$m_\Phi \times m_X \approx k(L + \log k)(L + \log d).$$

4. CONCLUSION

We have presented a very simply algorithm which allows to identify the k-active coordinates of a d-dimensional function using approximately $k \log k \log d$ function evaluations. To compare our result to [3], we need to take into account that once those coordinates are identified, we need another $\mathcal{O}(h^{-k})$ samples to identify g with precision $\|g\|_{\mathrm{Lip}} h$, where $\|g\|_{\mathrm{Lip}}$ is again the Lipschitz constant of g. Therefore, the actual number of samples needed to approximate f is given as the *sum* of (7) and $\mathcal{O}(h^{-k})$. This may be compared with the bound of [3], which involves the *product* of $\log_2 d$ and h^{-k}. Also, we avoid the pessimistic factor e^{k+1}. On the other hand, the constants implicitly involved in (7) are rather large, the conditions on g stronger and more complicated and the result holds only with high probability.

Our method is based on the numerical evaluation of the directional derivatives of f, as described in (2), which becomes unstable if the effective step size $\epsilon/\sqrt{m_\Phi}$ is chosen too small. However from (11) we see that we only require this effective size to be of the order of $k^{-3/2}$. In particular it does not depend on the dimension d.

Finally we want to mention that we expect the scheme to work even better, i.e. with significantly better constants, when measuring the size of the rows of $\Phi^T Y$ with the Euclidean instead of the maximum norm. This is work in progress to be found in the forthcoming paper [7].

5. REFERENCES

[1] E. Novak and H. Woźniakowski, "Approximation of infinitely differentiable multivariate functions is intractable," *Journal of Complexity*, vol. 25, pp. 398–404, 2009.

[2] E. Novak and H. Woźniakowski, *Tractability of Multivariate Problems, Volume I: Linear Information*, EMS Tracts in Mathematics, Vol. 6. Eur. Math. Soc., Zürich, 2008.

[3] R. A. DeVore, G. Petrova, and P. Wojtaszczyk, "Approximation of functions of few variables in high dimensions," *Constr. Approx.*, to appear.

[4] A. Cohen, I. Daubechies, R. A. DeVore, G. Kerkyacharian, and D. Picard, "Capturing ridge functions in high dimensions from point queries," *preprint*, 2010.

[5] M. Fornasier, K. Schnass, and J. Vybíral, "Learning functions of few arbitrary linear parameters in high dimensions," *preprint*, 2010.

[6] H. Rauhut, K. Schnass, and P. Vandergheynst, "Compressed sensing and redundant dictionaries," *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2210–2219, 2008.

[7] K. Schnass, "A simple algorithm to approximate functions of few variables in high dimensions," *in preparation*, 2011.